

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou

Faculté de : Génie électrique et d'informatique
Département : Informatique



Mémoire de fin d'études

En vue de l'obtention du diplôme de
Master en Informatique

Spécialité : Systèmes Informatiques

Présenté par :

AZZI Souad et **BOUDJEMAI Lamia**

Sujet : Mise en œuvre de l'apprentissage profond
pour la détection de Fake News

Proposé et dirigé par : SADI Samy

Soutenu le 26 septembre 2018 devant le jury composé de:

Mr. HABET Mohammed Saïd

Président du Jury

Mme. BOUGCHICHE Lilia

Membre du Jury

Mr. SADI Samy

Directeur de mémoire

Dédicaces

À mes très chers parents.

À mes sœurs et mes frères.

À mes nièces et mon neveu.

À tous mes amis et à tous mes camarades.

Souad

Dédicaces

À mes très chers parents.

À ma sœur et mes frères.

A tous mes amis et a tous mes camarades.

Lamia

Remerciements

Nous tenons d'abord à remercier très chaleureusement Monsieur **Samy SADI** qui nous a permis de bénéficier de son encadrement. Les conseils qu'il nous a prodigués, la patience et la confiance qu'il nous a témoignés ont été déterminants dans la réalisation de notre travail de recherche.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre modeste travail et de l'enrichir par leurs propositions.

Nos remerciements s'étendent également à tous nos enseignants durant nos années d'études.

Merci à Tous et à Toutes.

Résumé

Le terme Fake News (ou fausses informations) désigne des informations qui souvent sont volontairement truquées. Les Fake News ont toujours existées et ont aujourd'hui beaucoup d'impact à cause notamment de la généralisation des réseaux sociaux. Sur ces réseaux tout le monde peut diffuser une information sans obligation de la vérifier. Ainsi, les Fake news sont rapidement devenu un problème majeur sur l'internet. Pour faire face à la charge croissante de ce type d'information, plusieurs techniques de détection ont vu le jour. Dans ce mémoire, nous nous intéressons aux techniques à base de l'apprentissage automatique, mais plus particulièrement celles basées sur les réseaux de neurones. Pour classifier les news, nous construisons un classifieur à base du module d'instruction de phrase préformé nlm-en-128 et nous l'entraînons sur une collection de test préexistante Liar.

Mots clés: Fake News, apprentissage automatique, réseaux de neurones, classification, nlm-en-dim128, Liar.

Tables de matières

INTRODUCTION GENERALE.....	11
1	CONTEXTE DU TRAVAIL..... 12
2	PROBLEMATIQUE..... 12
3	CONTRIBUTIONS..... 13
4	ORGANISATION DU MEMOIRE 13
CHAPITRE 1: APPRENTISSAGE AUTOMATIQUE A BASE DE RESEAUX DE NEURONES	14
1	INTRODUCTION..... 15
2	HISTORIQUE 15
3	APPRENTISSAGE AUTOMATIQUE : QUELQUES DEFINITIONS..... 16
4	TYPES D'APPRENTISSAGE AUTOMATIQUE 16
4.1	<i>Apprentissage supervisé</i> 17
4.2	<i>Apprentissage non supervisé</i> 18
4.3	<i>Apprentissage semi-supervisé</i> 18
4.4	<i>Apprentissage partiellement supervisé</i> 18
4.5	<i>Apprentissage par renforcement</i> 18
4.6	<i>Apprentissage par transfert</i> 19
5	MODELE D'APPRENTISSAGE AUTOMATIQUE 19
5.1	<i>K plus proches voisins</i> 19
5.2	<i>Les arbres de décision</i> 20
5.3	<i>Les machines à vecteurs de support</i> 20
5.4	<i>Les réseaux de neurones</i> 21
6	MODELE FORMEL D'UN RESEAU DE NEURONES ARTIFICIEL 21
6.1	<i>Neurone biologique</i> 21
6.2	<i>Neurone formel</i> 22
6.3	<i>Réseau de neurone</i> 23
6.4	<i>Fonctionnement</i> 24
6.5	<i>Les fonctions d'activations usuelles</i> 25
6.5.1	<i>Fonction binaire à seuil</i> 26
6.5.2	<i>Fonction non linéaire</i> 26
6.5.3	<i>Fonction Gaussienne</i> 27
6.5.4	<i>Fonction Relu</i> 27
6.6	<i>Architecture des réseaux de neurones</i> 28
6.6.1	<i>Réseaux de neurones non bouclés</i> 28
6.6.2	<i>Réseaux de neurones bouclés</i> 29
7	ENTRAINEMENT D'UN RESEAU DE NEURONES ARTIFICIELS 30
7.1	<i>Le perceptron (sans rétro-propagation)</i> 31
7.2	<i>La rétro-propagation</i> 31
8	TOPOLOGIES D'UN RESEAU DE NEURONES ARTIFICIELS ET APPRENTISSAGE PROFOND : 32
8.1	<i>Les réseaux de neurones convolutifs</i> 32

8.2	<i>Les réseaux de neurones récurrents</i>	33
9	DEFIS RELATIFS A L'UTILISATION DES RESEAUX DE NEURONES.....	33
9.1	<i>La classification</i>	33
9.2	<i>La prédiction</i>	34
9.3	<i>La reconnaissance des formes</i>	34
9.4	<i>L'optimisation</i>	34
10	CONCLUSION.....	34
CHAPITRE 2: CLASSIFICATION DE TEXTES		35
1	INTRODUCTION.....	36
2	DEFINITION	36
3	PROCESSUS DE LA CATEGORISATION DE TEXTES.....	37
3.1	<i>Prétraitement du texte</i>	38
3.1.1	Lemmatisation et Racinisation	38
3.1.2	Mots vides	38
3.1.3	Nettoyage des mots	39
3.2	<i>La représentation des textes</i>	40
3.2.1	Représentation en sac de mots (<i>Bag of Words</i>)	40
3.2.2	Représentation par phrases	41
3.2.3	Représentation avec les racines lexicales	42
3.2.4	Représentation avec les lemmes	42
3.2.5	Représentation avec les n-grammes.....	42
3.2.6	Représentation conceptuelle	42
3.2.7	Représentation dans les réseaux de neurones	43
4	TYPES DE CLASSIFIEURS UTILISES.....	43
4.1	<i>Les machines à vecteurs de support</i>	43
4.2	<i>Les classifieurs naïfs de Bayes</i>	44
4.3	<i>Les classifieurs à base de réseaux de neurones</i>	45
5	LES APPLICATIONS DE LA CLASSIFICATION DES TEXTES	45
6	DEFIS DE LA CATEGORISATION DE TEXTES	45
7	CONCLUSION.....	47
CHAPITRE 3: LES FAKE NEWS		48
1	INTRODUCTION.....	49
2	DEFINITION	49
3	OBJECTIFS DE LEUR PUBLICATION	50
4	REPRESENTIONS DES FAKE NEWS.....	50
4.1	<i>NNLM</i>	51
4.2	<i>Représentation des métadonnées</i>	52
5	TOPOLOGIE DE NOTRE RESEAU DE NEURONES	53
6	CONCLUSION	53

CHAPITRE 4: IMPLEMENTATION	54
1 INTRODUCTION.....	55
2 LOGICIELS ET LIBRAIRIES UTILISES.....	55
2.1 <i>Logiciels</i>	55
2.1.1 PyCharm.....	55
2.1.2 ConEmu :.....	56
2.2 <i>Librairies</i>	56
2.2.1 Tensorflow	56
2.2.2 Tensorflow-Hub [30] [35]	57
2.2.3 Pandas	57
2.2.4 Numpy.....	57
2.2.5 Tensorboard.....	57
2.2.6 Python	58
3 PRESENTATION DE L'IMPLEMENTATION	58
4 ÉVALUATION.....	60
4.1 <i>Collection de test</i>	60
4.2 <i>Résultats</i>	61
4.2.1 Modèle 1 : Classification à base de contenu de la News et son thème	61
4.2.2 Modèle 2 : Classification à base de contenu de la News, son thème et l'auteur	62
4.2.3 Classification à base de contenu du contenu de la News, son thème, l'auteur et son affiliation	63
4.3 <i>Discussion</i>	64
5 CONCLUSION.....	65
CONCLUSION GENERALE	66
1 SYNTHÈSE	67
2 PERSPECTIVES	67
BIBLIOGRAPHIE	68

Table des figures

FIGURE 1: PRINCIPE DE L'APPRENTISSAGE SUPERVISE.....	17
FIGURE 2: EXEMPLE D'ARBRE DE DECISION	20
FIGURE 3: NEURONE BIOLOGIQUE.....	22
FIGURE 4:NEURONE FORMEL.....	23
FIGURE 5: EXEMPLE D'UN RESEAU DE NEURONE.....	24
FIGURE 6: NEURONE FORMEL.....	25
FIGURE 7: FONCTION SIGMOÏDE LOGARITHMIQUE.....	26
FIGURE 8 : FONCTION SIGMOÏDE TANGENTIELLE.....	27
FIGURE 9 : FONCTION GAUSSIENNE	27
FIGURE 10 : FONCTION RELU	28
FIGURE 11 : RESEAU DE NEURONES MULTICOUCHE.....	29
FIGURE 12 : LA FORWARD-PROPAGATION.....	31
FIGURE 13 : LA BACKWARD-PROPAGATION	32
FIGURE 14 : PROCESSUS DE CATEGORISATION.	37
FIGURE 15 : EXEMPLE D'UN TEXTE ET DE SON VECTEUR ASSOCIE. LES COMPOSANTES DU VECTEUR SONT SIMPLEMENT LES OCCURRENCES DES MOTS DU TEXTE.	41
FIGURE 16: LA SEPARATION DU L'HYPER PLAN PAR LES SVM.....	44
FIGURE 17: LES VECTEURS DE SUPPORT.....	44
FIGURE 18 : ENVIRONNEMENT PYCHARM.....	56
FIGURE 19 : TENSORBOARD	58
FIGURE 20 : FRAGMENT DE CODE.....	60
FIGURE 21 : PRECISION (ACCURACY) ET ERREUR (LOSS) AVEC LE MODELE 1 EN FONCTION DU NOMBRE D'ITERATIONS LORS DE L'ENTRAINEMENT.....	62
FIGURE 22 : PRECISION (ACCURACY) ET ERREUR (LOSS) AVEC LE MODELE 2 EN FONCTION DU NOMBRE D'ITERATIONS LORS DE L'ENTRAINEMENT.....	63
FIGURE 23 : PRECISION (ACCURACY) ET ERREUR (LOSS) AVEC LE MODELE 3 EN FONCTION DU NOMBRE D'ITERATIONS LORS DE L'ENTRAINEMENT.....	64

Table des tableaux

TABLEAU 1 : QUELQUES CARACTERISTIQUES FONCTIONNELLES ET LEURS TYPES DE RNA.....	33
TABLEAU 2 : PRECISIONS FINALES AVEC L'ENSEMBLE D'ENTRAINEMENT ET L'ENSEMBLE DE TEST AVEC LE MODELE 1.....	61
TABLEAU 3 : PRECISIONS FINALES AVEC L'ENSEMBLE D'ENTRAINEMENT ET L'ENSEMBLE DE TEST AVEC LE MODELE 2.....	62
TABLEAU 4 : PRECISIONS FINALES AVEC L'ENSEMBLE D'ENTRAINEMENT ET L'ENSEMBLE DE TEST AVEC LE MODELE 3.....	63
TABLEAU 5 : TABLEAU COMPARATIFS DE RESULTATS.....	64

Introduction générale

1 Contexte du travail

Ces dernières années, les Fake News ont pris une ampleur médiatique importante, surtout au cours des élections présidentielles de 2016 aux Etats-Unis. Ce phénomène se propage dans des buts différents soit pour tromper le lecteur ou pour influencer son opinion sur des sujets particuliers. Il est donc important de disposer d'outils nous permettant de détecter ce genre de fausses nouvelles pour s'en prémunir, notamment grâce à l'intelligence artificielle.

L'Intelligence Artificielle (IA) est l'un des domaines les plus récents de la science et de l'ingénierie. Pourtant, des recherches s'y rapportant ont commencées dès la seconde guerre mondiale et le nom lui-même a été inventé en 1956. L'IA englobe plusieurs sous domaines allant du plus général (apprentissage et perception) au plus spécifique comme jouer aux échecs, démontrer des théorèmes mathématiques, écrire des poèmes, détecter des Spams (courriels indésirables), conduire une voiture, diagnostiquer des maladies ou encore détecter des Fake News. L'IA se révèle être utile dans toutes les tâches intellectuelles. Le but de la recherche en IA est de créer une technologie qui permette aux ordinateurs et aux machines de fonctionner d'une manière intelligente.

La recherche en apprentissage automatique a produit une large gamme d'algorithmes supervisés pour construire des classifieurs. Dans ce mémoire, nous nous intéressons aux approches à base des réseaux de neurones. Plus particulièrement, nous nous intéressons aux approches orientées vers l'analyse de textes, pour les appliquer à la détection de Fake News.

2 Problématique

Il existe plusieurs problématiques relatives à la détection de Fake News. C'est une tâche de classification compliquée et nous pouvons résumer les problématiques en ce qui suit.

Tout d'abord, il faut bien définir ce qu'est une Fake News. En effet, est-ce que l'énoncé même de la nouvelle suffit à définir la News ? Pouvons-nous prendre en compte d'autres paramètres ou métadonnées relatives à la News lors de la classification pour éventuellement améliorer les résultats de classification.

Ensuite, en second lieu, il faut déterminer comment sera implémenté le classifieur. En effet, il faut bien comprendre et exploiter les outils existants, pour construire un modèle de réseaux de neurones efficace permettant de classifier les News.

3 Contributions

Nos contributions peuvent être résumées en ce qui suit :

- Nous avons réalisé un état de l'art sur l'apprentissage automatique, mais aussi, sur la représentation et la classification de textes.
- Nous avons construit un modèle pour la classification de Fake News en se basant sur des modèles existants dans l'état de l'art que nous avons implémenté.
- Nous avons implémenté et testé le modèle construit sur une collection de test existante et conclu quant à l'utilité de l'IA pour la détection de Fake News.

4 Organisation du mémoire

Ce mémoire est organisé en 2 parties totalisant 4 chapitres en plus de l'introduction générale et de la conclusion générale.

La première partie compte le premier et deuxième chapitre. Nous y donnons un aperçu des recherches actuelles sur l'apprentissage automatique dans le chapitre. Ensuite, dans le deuxième chapitre, nous présentons la notion de catégorisation de textes, sa définition, ainsi que son processus.

La deuxième partie compte le troisième et quatrième chapitre. Nous détaillons nos contributions relatives au modèle utilisé et à notre implémentation. Ainsi, dans le troisième chapitre, nous décrivons le modèle utilisé pour la classification de Fake News. Ensuite, dans le quatrième chapitre, nous décrivons notre implémentation, les outils et bibliothèques utilisées et les résultats obtenus.

Chapitre 1: Apprentissage automatique à base de réseaux de neurones

1 Introduction

L'intelligence artificielle et l'apprentissage automatique sont des domaines connaissant un intérêt des plus accrus et en constante augmentation aussi bien de la part de communauté scientifique que de la part des entreprises. Les opportunités qu'offre l'apprentissage automatique qui permet, notamment, de construire des classifieurs performants de tous types de façon totalement automatisée, justifient cet intérêt.

L'apprentissage automatique est un sous domaine de l'intelligence artificielle qui s'intéresse au développement, à l'analyse et à l'implémentation de méthodes permettant à une machine d'évoluer grâce à un processus d'apprentissage et ainsi de remplir des tâches qu'il est difficile où impossible de remplir par des moyens algorithmiques classiques. L'objectif de l'apprentissage automatique est ainsi de concevoir des programmes pouvant s'améliorer automatiquement avec l'expérience.

Différentes méthodes et modèles peuvent être utilisés pour l'apprentissage automatique. L'une des approches les plus réputées consiste néanmoins en l'utilisation de réseaux de neurones. C'est ce sur quoi nous nous focalisons dans le cadre de notre travail.

Dans ce chapitre, nous présentons l'apprentissage automatique en donnant son historique et quelques définitions. Ensuite, nous présentons les différents types d'apprentissage automatique ainsi que ses différentes formes. Après quoi, nous nous intéressons aux réseaux de neurones. Nous exposons en détails leurs propriétés, leur fonctionnement et les défis relatifs à leur utilisation. Enfin, nous terminons ce chapitre avec une conclusion.

2 Historique

L'apprentissage automatique est une discipline jeune, à l'instar de l'intelligence artificielle, mais elle a déjà une histoire

La concrétisation de cette discipline est principalement due à Alan Turing et son concept de la « Machine universelle » en 1936 qui est à la base des ordinateurs d'aujourd'hui. [1]

En 1943, le neurophysiologiste Warren McCulloch et le mathématicien Walter Pitts publièrent un article décrivant le fonctionnement de neurones en les représentant à l'aide de circuits électriques. Cette représentation sera la base théorique des réseaux neuronaux.

Arthur Samuel, informaticien américain pionnier dans le secteur de l'intelligence artificielle, fut le premier à faire usage du terme « apprentissage automatique » (*Machine Learning* en anglais) en 1959 suite à la création de son programme pour IBM en 1952. Le programme jouait au jeu de Dames et s'améliorait plus il jouait. À terme, il parvint à battre le meilleur joueur des États-Unis.

Durant les années suivantes, les applications de l'apprentissage automatique médiatisées se succèdent bien plus rapidement qu'auparavant. En 2012, Un réseau neuronal développé par Google parvient à reconnaître des visages humains ainsi que des chats dans des vidéos YouTube.

En 2015, une nouvelle étape importante est atteinte, lorsque l'ordinateur Google « Alpha GO » gagne contre un des meilleurs joueurs au jeu de plateau considéré comme le plus dur du monde : le jeu de Go. En 2016, un système d'intelligence artificielle à base d'apprentissage automatique nommé « LipNet » parvient à lire les lèvres avec un grand taux de succès. [2].

3 Apprentissage Automatique : Quelques définitions

L'apprentissage automatique ou artificiel regroupe tellement de concepts différents et variés qu'il est difficile d'en donner une définition unique. En effet l'apprentissage automatique se situe au carrefour de plusieurs disciplines incluant : l'intelligence artificielle, l'analyse des données et les statistiques, la philosophie, la psychologie, la théorie de l'information, la biologie, les sciences cognitives, etc. Nous présentons les définitions suivantes parmi les plus proéminentes:

Définition 1 : L'apprentissage automatique ou « *Machine Learning* » est une méthode utilisée en intelligence artificielle. Il s'agit d'algorithmes (procédures traduites en langages informatiques) qui analysent un ensemble de données afin de déduire des règles qui constituent de nouvelles connaissances permettant d'analyser de nouvelles. [3]

Définition 2 : Cette notion englobe toute méthode permettant de construire un modèle de la réalité à partir de données, soit en améliorant un modèle partiel ou moins général, soit en créant complètement le modèle. Il existe deux tendances principales en apprentissage, celle issue de l'intelligence artificielle et qualifiée de symbolique, et celle issue des statistiques et qualifiée de numérique. [4]

Définition3 : L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques. [2]

4 Types d'apprentissage automatique

L'apprentissage automatique est un sujet central depuis le début de l'intelligence artificielle, il en existe de nombreuses approches. Elles sont regroupées en plusieurs familles qui se différencient par le type d'information dont dispose le système pour apprendre et le protocole avec lequel il interagit avec son environnement. Nous distinguons les types suivant :

- Apprentissage supervisé
- Apprentissage non-supervisé
- Apprentissage semi-supervisé
- Apprentissage partiellement supervisé
- Apprentissage par renforcement
- Apprentissage par transfert

4.1 Apprentissage supervisé

Il est parmi les méthodes d'apprentissage par induction consistant à construire automatiquement un classifieur qui apprend, à partir des exemples déjà classés (ou étiquetés). Ce type est dit supervisé parce que la fonction de classification s'entraîne sur les classes ainsi que sur leurs caractéristiques. Nous distinguons deux problèmes auxquels l'apprentissage supervisé est appliqué : la classification et la régression.

Définition de la classification : La classification cherche à prédire l'appartenance d'objets à des classes connues à priori en examinant leurs caractéristiques.

Définition d'un classifieur : Le modèle construit par un algorithme d'apprentissage supervisé est appelé classifieur. Ce dernier une fois élaboré doit être capable de classer les nouveaux exemples.

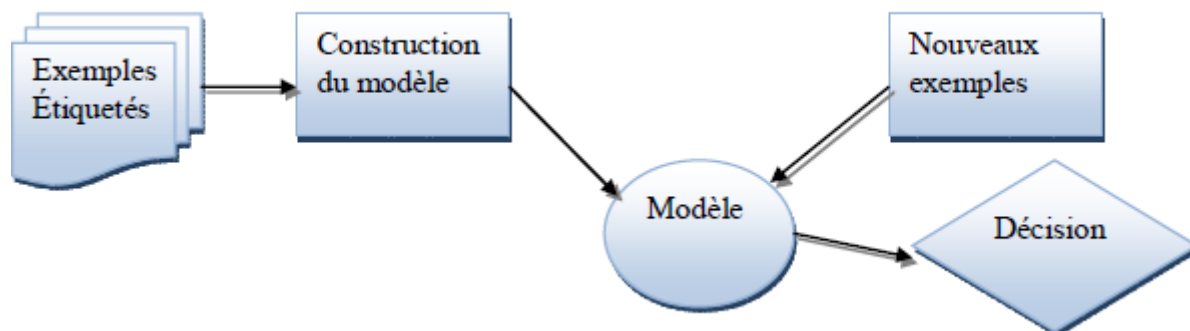


Figure 1: Principe de l'apprentissage supervisé.

Dans l'apprentissage supervisé, la construction et la génération d'un modèle passe par deux phases : la phase d'apprentissage et la phase de test.

La phase d'apprentissage (ou d'entraînement) : La phase d'apprentissage représente l'étape de construction et de génération d'un modèle grâce à un algorithme qui prend en entrée un ensemble d'exemples de données qui contient toutes les informations nécessaires pour caractériser un problème donné.

La phase de test : Dans la phase de test le modèle généré doit être capable de prédire l'étiquette d'un nouvel exemple en fonction des valeurs d'entrées.

4.2 Apprentissage non supervisé

L'apprentissage non supervisé est aussi appelé apprentissage sans enseignant. Dans ce type d'apprentissage, la tâche consistera en la découverte de similarités entre les observations (entrées) dans une collection d'exemples, dans le but de regrouper celles-ci en sous-ensembles appelés clusters ou classes. Une technique employée consiste à implémenter des algorithmes pour rapprocher les exemples les plus similaires et éloigner ceux qui ont le moins de caractéristiques communes.

Aucun expert n'est disponible pour ce type d'apprentissage, il se fait de manière totalement autonome sans l'aide d'un superviseur. L'objectif de cette technique est de procéder à une classification du type regroupement par similitude.

Parmi les algorithmes d'apprentissage non supervisé, nous citons : la k-moyenne (k-Means : recherche des plus proches voisins), le clustering hiérarchique, etc.

4.3 Apprentissage semi-supervisé

Il utilise un ensemble de données étiquetées et non-étiquetées. Il se situe ainsi entre l'apprentissage supervisé qui n'utilise que des données étiquetées et l'apprentissage non-supervisé qui n'utilise que des données non-étiquetées. L'utilisation de données non étiquetées en combinaison avec des données étiquetées, permet d'améliorer de façon significative la qualité de l'apprentissage.

4.4 Apprentissage partiellement supervisé

Dans ce type d'apprentissage, les classes sont connues partiellement, c'est à dire nous disposons d'informations partielles concernant les classes associées aux exemples d'apprentissage.

L'apprentissage se fait à partir d'une séquence d'observation partiellement supervisée. Chaque observation est associée à une ou plusieurs étiquettes dont une de ces derniers est la bonne. C'est le cas quand un modèle énonce que des données appartient à une classe A , B ou C et d'autres n'appartient pas à A mais peut-être à une classe B ou C

4.5 Apprentissage par renforcement

L'apprentissage par renforcement [5] est une approche qui vise à résoudre des problèmes complexes où nous n'indiquons au système que le but à atteindre et ce dernier doit apprendre par une succession d'essais et d'erreurs, comment atteindre le but fixé.

L'apprentissage par renforcement fait référence donc à une classe de problèmes d'apprentissage automatique, dont le but est d'apprendre, à partir d'expériences, ce qu'il convient de faire en différentes situations, de façon à optimiser une récompense numérique au cours du temps. Il s'agit

d'une évaluation non assistée qui fournit une mesure relative de la performance. Le système doit identifier les actions à exécuter pour que cette mesure soit maximisée. L'apprentissage par renforcement consiste à apprendre quoi faire, comment associer des actions à des situations, afin de maximiser quantitativement une récompense, dite aussi le signal de renforcement. L'algorithme de Q-Learning est un exemple classique

4.6 Apprentissage par transfert

L'apprentissage par transfert [6] peut être vu comme la capacité d'un système à reconnaître et appliquer des connaissances et des compétences, apprises à partir de tâches antérieures, sur de nouvelles tâches ou domaines partageant des similitudes.

C'est une méthode de l'apprentissage automatique où un modèle développé pour une tâche est réutilisé comme point de départ pour un modèle sur une deuxième tâche.

Il consiste à utiliser un jeu de tâches pour influencer l'apprentissage et améliorer les performances sur une autre tâche.

5 Modèle d'apprentissage automatique

Malgré la jeunesse du domaine de l'apprentissage automatique, il existe beaucoup d'algorithmes d'apprentissage différents, chacun possédant ses propres avantages.

Dans ce qui suit, nous dériverons quelques algorithmes de l'apprentissage tels que :

- K plus proches voisins
- les arbres de décision
- Les machines à vecteurs de support
- Réseaux de neurones

5.1 K plus proches voisins

Souvent abrégée en KPPV en français, la méthode des K plus proche voisins [7] est simple mais efficace dans beaucoup de cas. Le principe est d'assigner à la donnée d'entrée la classe majoritaire parmi ses plus proches voisins dans l'échantillon d'apprentissage.

Pour prédire la classe d'une nouvelle donnée, l'algorithme calcule sa similarité avec toutes les données déjà classées. Puis, il trie les valeurs de similarité par ordre décroissant et ne garde que les K premières. La classe la plus fréquente parmi les K données restantes est celle qui est attribuée à la nouvelle donnée.

Donc, comme son nom l'indique la classification d'une nouvelle donnée dépend de ses K voisins les plus proches.

5.2 Les arbres de décision

Les arbres de décision [7] sont parmi les techniques les plus populaires de l'apprentissage automatique, il se situe dans le cadre de l'apprentissage supervisé. L'arbre de décision comme son nom l'indique est un regroupement de fonctions locales structurés en forme d'arbre.

Un classifieur de type arbre de décision repose sur un arbre où chaque nœud interne est marqué par un terme et est associé à une condition sur ce terme. Les branches sortant de nœud correspondent aux résultats de test et les feuilles sont étiquetées par les catégories.

Donc, l'arbre de décision est une représentation visuelle d'un algorithme de classification de données suivant différents critères que nous désignons par décisions(ou nœuds).

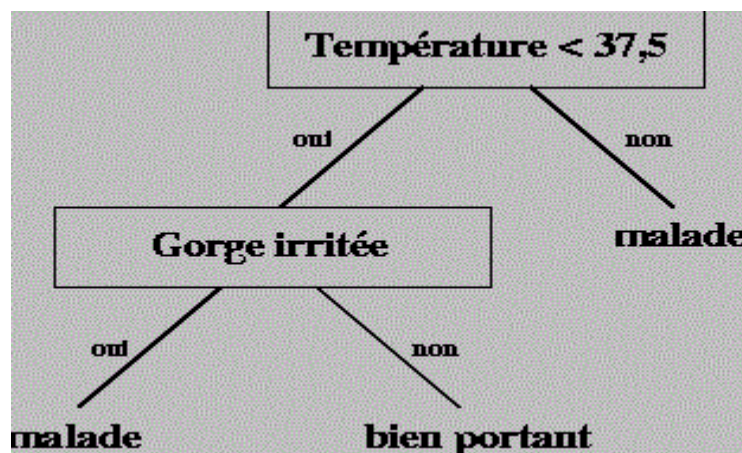


Figure 2: Exemple d'arbre de décision

5.3 Les machines à vecteurs de support

Les machines à vecteur de support [8] [9] (SVM ou bien « *Support Vector Machines* ») ont été développées par Vapnick, et sont des un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes des classifications et de régression.

L'idée principale de cet algorithme est de trouver un hyperplan qui sépare au mieux les exemples positifs des exemples négatifs en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale.

Les SVMs ont rapidement été adoptées pour leur capacité à travailler avec des données de grandes dimensions, elles s'appartiennent aux classifieurs linéaires dans le sens où elles tentent de séparer l'espace en deux mais certaines manipulations mathématiques les rendant adaptables à des problèmes non linéaires.

Les SVMs ont été appliqués à de très nombreux domaines tels que la bioinformatique, la recherche d'information, la finance, etc.

5.4 Les réseaux de neurones

Introduit pour la première fois par McCulloch et Pitts 1943, les réseaux de neurones [1] (parfois appelés réseaux connexionnistes) sont des piliers de l'intelligence artificielle et de l'apprentissage automatique. Les principales applications des réseaux de neurones sont l'optimisation et l'apprentissage. En apprentissage, les réseaux de neurones sont essentiellement utilisés pour l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

Un réseau de neurones est formé d'un ensemble d'unités de traitement (neurones) interconnectées par des liaisons pondérées, ces réseaux sont organisés selon différentes topologies, la plus courante étant le perceptron multicouche. L'entraînement d'un réseau de neurones consiste à modifier de façon dynamique les pondérations appliquées aux différentes liaisons selon un algorithme conçu pour minimiser l'erreur finale de prédiction.

Les réseaux de neurones ont été appliqués à de très nombreux domaines : le biomédical, la bioinformatique, la défense, les télécommunications, etc.

Nous présentons plus en détail ce modèle d'apprentissage automatique dans ce qui suit.

6 Modèle formel d'un réseau de neurones artificiel

6.1 Neurone biologique

En biologie [39], le cerveau humain contient un grand nombre de neurones fortement interconnectés constituant des réseaux de neurones. Un neurone est une cellule nerveuse constituant la base du système nerveux spécialisée dans le traitement des signaux électriques. Chaque neurone est une entité autonome au sein du cerveau. Il est constitué de :

- Corps cellulaire (unité principale)
- Dendrites (organes d'entrée)
- Axone (organe de sortie)
- Synapses (sorties)

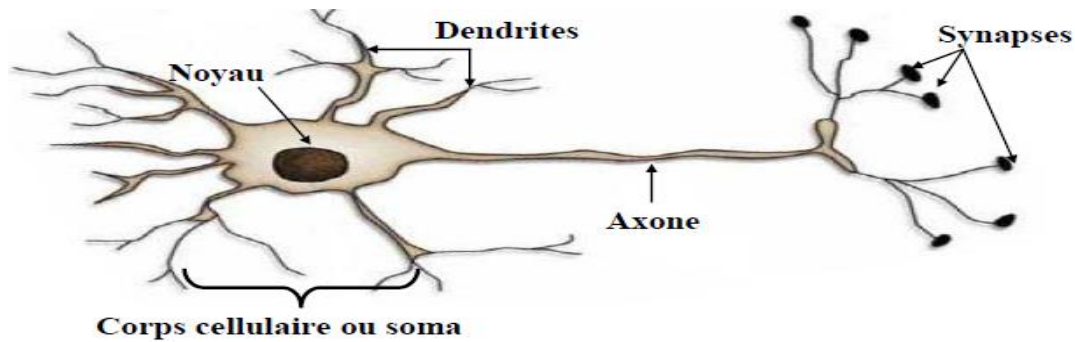


Figure 3: Neurone biologique.

6.2 Neurone formel

Définition 1 : Un neurone formel (artificiel) est une fonction de traitement qui reçoit des données en entrée, sous la forme d'un vecteur, et produit en sortie une autre donnée. Cette sortie est une fonction des entrées et des poids de connexion.

Définition 2 : Un neurone formel ou artificiel, aussi fréquemment appelé neurone de McCulloch et Pitts, est un opérateur mathématique très simple qui traduit la représentation mathématiques d'un neurone biologique. Selon, G. Dreyfus : « un neurone est une fonction algébrique non linéaire, paramétrée, à valeur bornée ».

Il possède des entrées qui peuvent être les sorties d'autres neurones, ou des entrées de signaux extérieurs, et une sortie. La valeur de la sortie résulte du calcul de la somme des entrées, pondérées par des coefficients (dits **poids synaptiques**) et du calcul d'une fonction non linéaire bornée de cette somme pondérée (dite **fonction d'activation**).

Un neurone formel peut être défini par les cinq éléments suivants :

- **Entrée :** Il s'agit d'un vecteur qui comprend les grandeurs physiques à analyser;
- **Poids :** Les paramètres W_n sont souvent désignés sous le nom de «poids» ou, en raison de l'inspiration biologique des réseaux de neurones, «poids synaptiques». Ils sont des valeurs réelles qui déterminent la contribution de chaque entrée. Un algorithme d'apprentissage permet de déterminer le meilleur ensemble de poids pour le problème considéré. Trouver l'ensemble optimal est souvent un compromis entre temps de calcul, minimisation de l'erreur et conservation de la capacité de généralisation pour le réseau
- **Biais :** est parfois appelé seuil, est un nombre réel qui est soustrait de la somme pondérée des valeurs d'entrée. Pour des raisons de simplicité et sert à contrôler si un neurone est actif, le biais peut être vu comme une entrée supplémentaire associée à un poids.

- **Fonction d'activation :** La fonction d'activation est une fonction mathématique permet de définir l'état interne du neurone en fonction de son entrée total(les neurones calculent leurs état à partir d'information qu'ils reçoivent).
- **Sortie du neurone :** La sortie du neurone ni d'autre que l'application de la fonction de transfert à la somme pondérée de ces entrées. La sortie calculée, qui peut être discrète ou réelle selon la fonction d'activation utilisée, passe aux autres neurones ou directement à l'environnement extérieur.

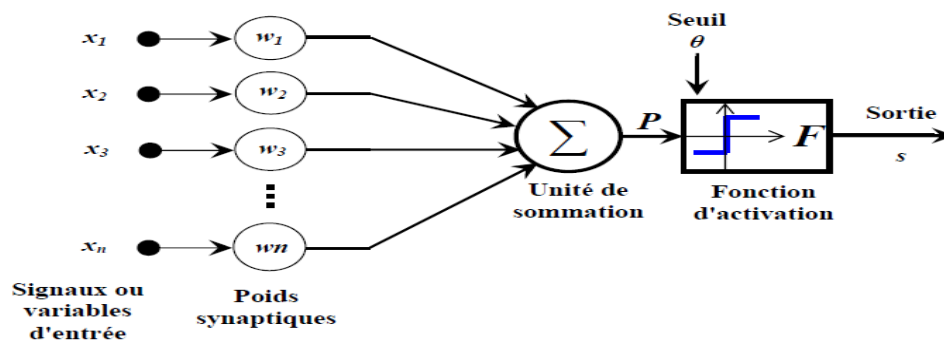


Figure 4:Neurone formel.

6.3 Réseau de neurone

Dans les ordinateurs traditionnels, l'information est mémorisée de façon statique et que les traitements sont effectués par des processeurs séquentiels. Dans l'approche connexionniste, l'information est stockée comme configurations d'un réseau constitué de très nombreux processeurs élémentaires connectés de façon dynamique.

Ces réseaux sont construits en s'inspirant de la physiologie et du fonctionnement du système nerveux. Ils sont constitués d'un grand nombre de neurones. Les connexions peuvent évoluer au cours de phases d'apprentissage. D'une telle auto-organisation peuvent émerger des conduites intelligentes qui n'ont pas été nécessairement définies par les concepteurs du système.

Définition 1: Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires (les neurones formels) fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit.

Définition 2 : Un réseau de neurones peut être considéré comme un modèle mathématique de traitement réparti, composé de plusieurs éléments de calcul non linéaires, opérant en parallèle et connectés entre eux par des poids. Un réseau de neurones permet le calcul des sorties (y_i), qui représentent l'approximation de la fonction désirée en fonction des entrées (x_i). La figure suivante montre un exemple d'un réseau de neurones multicouche.

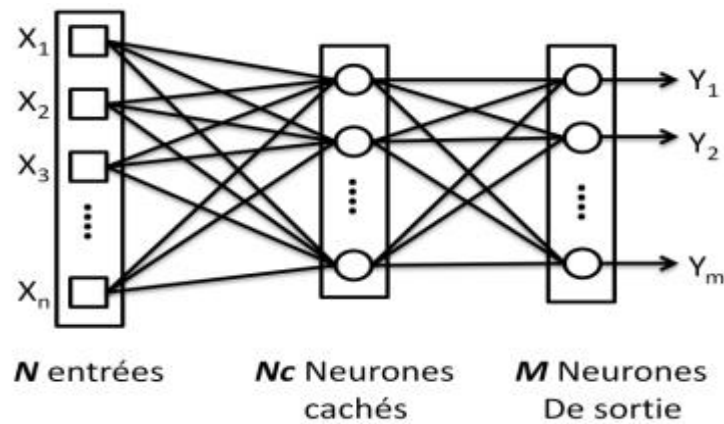


Figure 5: Exemple d'un réseau de neurone

L'information est portée par la valeur de ces poids, tandis que la structure du réseau de neurones ne sert qu'à traiter cette information et à l'acheminer vers la sortie. [38]

6.4 Fonctionnement

Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones en amont ou des capteurs composant la machine dont il fait partie. A chacune de ses entrées est associé un poids représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones en aval.

Le neurone réalise alors trois opérations sur ses entrées :

- **Pondération** : multiplication de chaque entrée par un paramètre appelé poids de connexion.
- **Sommation** : une sommation des entrées pondérées est effectuée, cette somme est appelée potentiel neuronal.
- **Activation** : passage de cette somme dans une fonction, appelée fonction d'activation fonction f appelée généralement fonction d'activation ou d'évaluation. La valeur calculée est la sortie du neurone qui est transmise aux neurones suivants.

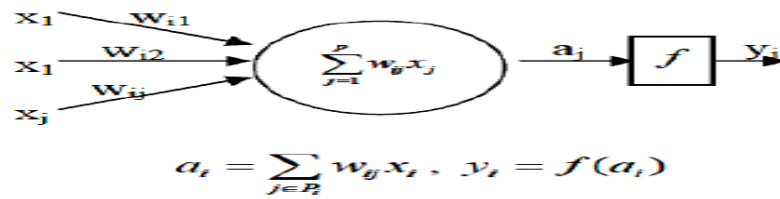


Figure 6: Neurone formel

La figure 6 présente les différents composants d'un neurone formel. x_i est le paramètre d'entrée. a_i est le potentiel neuronal (ou la sortie du neurone avant application de la fonction d'activation). w_{ij} est le poids d'une connexion. Enfin, y_i est la sortie du neurone. f est la fonction d'activation.

Une collection des neurones formels composent un réseau de neurones. Ces réseaux sont souvent constitués d'un nombre fini de neurones qui sont arrangés sous forme de couches connectés, d'une façon à ce que la sortie d'un neurone puisse être l'entrée d'un ou plusieurs d'autre neurone, par des poids de connexion. L'information dans le réseau se propage d'une couche à l'autre.

Nous distinguons trois types de couches de neurones :

- **Couche d'entrée** : les neurones de cette couche reçoivent les valeurs d'entrée du réseau et les transmettent aux neurones cachés. Chaque neurone reçoit une valeur, il ne fait pas donc de sommation.
- **Couches cachées** : chaque neurone de cette couche reçoit l'information de plusieurs couches précédentes, effectue la sommation pondérée par les poids, puis la transforme selon sa fonction d'activation qui est en général une fonction sigmoïde. Par la suite, il envoie cette réponse aux neurones de la couche suivante.
- **Couche de sortie** : elle joue le même rôle que les couches cachées, la seule différence entre ces deux types de couches est que la sortie des neurones de la couche de sortie n'est liée à aucun autre neurone.

6.5 Les fonctions d'activations usuelles

La fonction de transfert ou d'activation calcule la valeur de l'état du neurone. C'est cette valeur qui sera transmise aux neurones. Il existe de nombreuses formes possibles pour la fonction de transfert.

Nous décrivons dans ce qui suit, les formes les plus courantes des fonctions de transfert.

6.5.1 Fonction binaire à seuil

Elle a été utilisée par McCulloch et Pitts dans leur modèle du neurone formel. Elle fait passer l'activation du neurone d'une valeur à une autre dès que l'entrée résultante dépasse un certain seuil. Le seuillage introduit une non linéarité dans le comportement du neurone. Cependant, il limite la gamme de ses réponses possibles à deux valeurs.

6.5.2 Fonction non linéaire

6.5.2.1 Fonction sigmoïde logarithmique

La fonction sigmoïde est l'équivalent continu de la fonction linéaire bornée. Vu ses nombreux avantages (continuité et différentiabilité), ce type de fonction est généralement très employé dans le réseau de neurones.

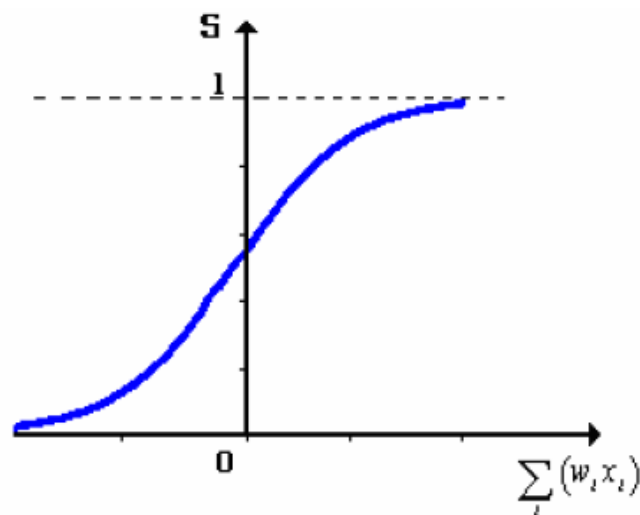


Figure 7: Fonction sigmoïde logarithmique.

6.5.2.2 La fonction sigmoïde tangentielle

La fonction sigmoïde tangentielle ou tangente hyperbolique est fortement utilisée dans les réseaux de neurones multicouches (qui possèdent une ou plusieurs couches de neurones cachées).

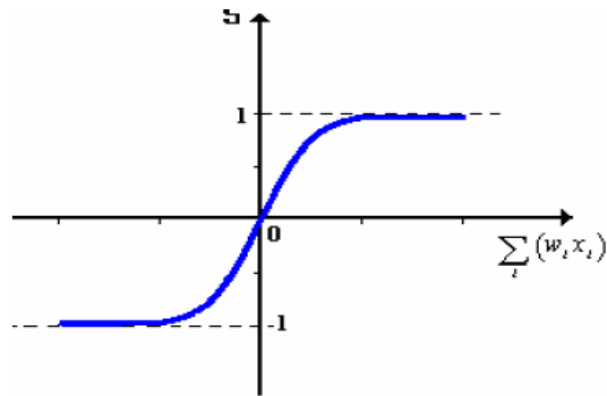


Figure 8 : Fonction sigmoïde tangentielle

6.5.3 Fonction Gaussienne

Cette fonction n'est pas monotone, elle est souvent employée dans les réseaux RBF (Radial Basis Function) (présenté plus loin dans ce chapitre).

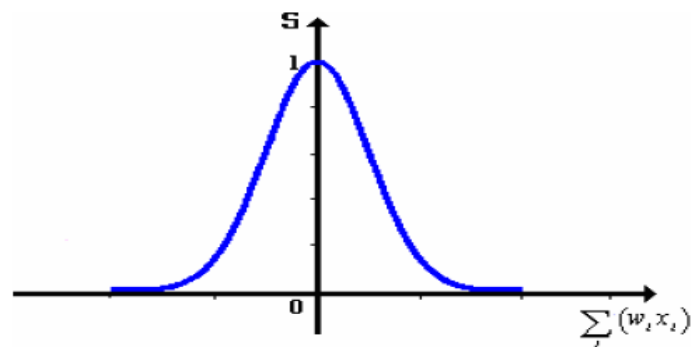


Figure 9 : Fonction Gaussienne

6.5.4 Fonction Relu

Relu (Rectified Linear Units) désigne la fonction réelle non-linéaire définie par :

$$Relu(x) = \max(0, x)$$

Cette fonction remplace toutes les valeurs négatives reçues en entrées par des zéros.

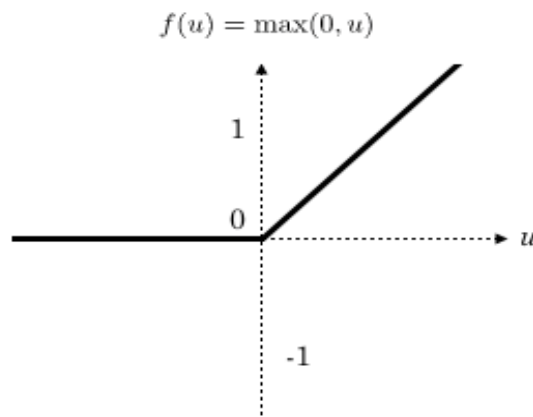


Figure 10 : Fonction Relu

6.6 Architecture des réseaux de neurones

Sur le plan architectural, un réseau de neurones peut être vu comme un ensemble d'unités élémentaires (les neurones) interconnectées de manière à constituer un système avec une ou plusieurs entrées et une ou plusieurs sorties. Modéliser un réseau de neurones, c'est décrire le modèle du neurone et le modèle des connexions entre ces neurones. Selon le type de connexion, nous distinguons deux types de réseaux de neurones : les réseaux bouclés et non bouclés.

6.6.1 Réseaux de neurones non bouclés

Dans un réseau de neurones non bouclés [40] (*Feed-Forward Neural Network*), le flux d'information circule des entrées vers les sorties sans « retour en arrière ». Selon le nombre de couches dans un réseau et la fonction d'activation, nous distinguons :

6.6.1.1 Le Perceptron

C'est l'un des premiers réseaux de neurones, conçu en 1958 par F. Rosenblatt. Il est linéaire et représente l'unité élémentaire d'un réseau de neurones. Le perceptron se compose de deux couches : la couche d'entrée et la couche de sortie qui donne la réponse correspondant à la simulation présente en entrée.

6.6.1.2 Le Perceptron Multicouche (MLP)

Le Perceptron multicouche (MLP : *Multi Layer Perceptron*) est un réseau à plusieurs couches cachées entre les couches d'entrée et de sortie. Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante et il n'y a pas de connexions entre les neurones d'une même couche. Le MLP a été appliqué avec succès dans des problèmes non-linéairement séparables et des problèmes logique plus compliqués.

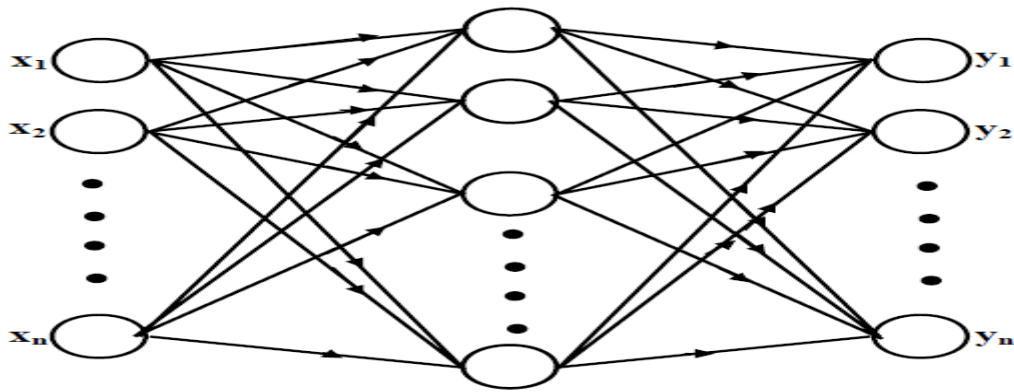


Figure 11 : Réseau de neurones multicouche.

6.6.1.3 Réseaux RBF (Radial Basis Function)

Ces réseaux ont été introduits par *M. Powel* et *D. Broomhead*. Il s'agit d'une "spécialisation" d'un perceptron multicouche (MLP). Les neurones de ces réseaux sont actifs pour une zone restreinte de l'espace d'entrée qui est définie par des fonctions radiales. Cette technique consiste à utiliser la non linéarité d'une fonction (par exemple une gaussienne) pour ponctuer la réponse des neurones des couches cachées.

6.6.2 Réseaux de neurones bouclés

Il existe plusieurs types de réseaux de neurones bouclés [28]. Le point commun entre tous ces réseaux est la présence d'au moins un chemin qui part et revient au même neurone.

6.6.2.1 Modèle de Hopfield

Le modèle élaboré par J. Hopfield, est totalement interconnecté c'est le réseau bouclé par excellence. Il n'y a plus de notions de couche, du fait que chaque neurone du réseau est à la fois une entrée et une sortie, le réseau implantant ainsi une mémoire associative (adressables par le contenu), c'est-à-dire que nous lui donnons des exemples et il doit être capable de les reconnaître lorsqu'ils sont bruités. Ce type de réseau est utilisé essentiellement dans la reconnaissance de la parole et la reconnaissance des formes.

6.6.2.2 Modèle de Boltzmann

La Machine de Boltzmann qui a été introduite en 1984 par G. E. Hinton, est le premier modèle connu apte à traiter de manière satisfaisante les limitations recensées dans le cas du perceptron. Il s'agit d'un réseau fortement récurrent au même titre que les réseaux de Hopfield. Mais à la différence de ces derniers, le réseau est divisé en trois parties distinctes et les entrées et sorties sont clairement séparées et identifiées. L'utilisation pratique de ces réseaux s'avère difficile, la convergence de l'algorithme étant extrêmement longue (les temps de calcul sont considérables). Il est souvent employé dans le domaine de reconnaissance de la parole ou de la vision.

6.6.2.3 Modèle de Kohonen

La carte de Kohonen est un modèle de réseaux de neurones, en général à deux dimensions. Chaque neurone de la couche d'entrées est relié à chaque neurone de la carte de Kohonen. Chaque neurone de la carte de Kohonen est relié à tous les neurones de la carte. Les poids sont déterminés en fonction de la distance selon une fonction « chapeau mexicain » ou DOG (Difference of Gaussians). Ces réseaux fonctionnent dans leur très grande majorité dans un mode non-supervisé.

7 Entraînement d'un réseau de neurones artificiels

L'entraînement d'un RNA [10] (réseau de neurones artificiels) passe par l'utilisation d'un ensemble d'équations mathématiques, afin « d'apprendre par l'expérience ». Pour qu'un réseau fonctionne, nous lui fournissons des données en entrée qu'il doit « apprendre » à reconnaître, soit pour les classer en sous-groupes homogènes (apprentissage non-supervisé), soit pour les associer à une variable indicatrice de sortie (apprentissage supervisé, conformément aux objectifs du présent mémoire). Nous adoptons la définition suivante: « L'apprentissage est un processus dynamique et itératif permettant de modifier les paramètres d'un réseau en réaction avec les stimuli qu'il reçoit de son environnement. Le type d'apprentissage est déterminé par la manière dont les changements de paramètre surviennent ». Il y a donc plusieurs types d'apprentissage que nous pouvons appliquer dans un réseau de neurones, mais nous nous intéressons qu'à un seul qui sera utilisé par la suite dans ce travail (i.e. l'apprentissage supervisé). Dans ce type d'apprentissage, le réseau s'adapte par comparaison entre le résultat qu'il a calculé, en fonction des entrées fournies et la réponse attendue en sortie. Ainsi, le réseau va se modifier jusqu'à ce qu'il trouve la bonne sortie, c'est-à-dire celle attendue, correspondant à une entrée donnée.

Un réseau de neurones est un maillage de plusieurs neurones, généralement organisé en couches. Pour construire une couche de S neurones, il s'agit simplement de les assembler. Les S neurones d'une même couche sont tous branchés aux R entrées. La couche est dite alors totalement connectée, un poids w_{ij} est associé à chacune des connexions. La dernière couche est nommée «couche de sortie». Les couches qui précèdent la couche de sortie sont nommées «couches cachées». Pour spécifier la structure du réseau, il faut aussi choisir le nombre de couches et le nombre de neurones sur chaque couche.

Entraîner un réseau de neurones signifie modifier la valeur de ses poids et de ses biais pour qu'il réalise la fonction d'entrée-sortie désirée. Nous disposons d'exemples d'apprentissage étiquetés, c'est à dire de couple (x, y) où x est le vecteur d'entrée présenté au réseau et y le vecteur de sortie que nous souhaitons obtenir.

7.1 Le perceptron (sans rétro-propagation)

Un réseau de sans propagation se compose de plusieurs couches successives de neurones avec une couche d'entrée, plusieurs couches cachées et une couche de sortie.

7.2 La rétro-propagation

La rétro-propagation (en anglais *back propagation*) est une méthode de calcul des poids qui consiste à minimiser l'erreur quadratique de sortie (somme des carrés de l'erreur de chaque composante entre la sortie réelle (O_p) et la sortie désirée (Y_p)).

L'algorithme de la rétro-propagation s'applique en deux étapes :

- la Forwad-propagation : durant laquelle les entrées (X_i) sont appliquées à la couche d'entrée et se propagent, en avant, dans le réseau pour calculer : la sortie O_p et l'erreur ($Y_p - O_p$) par rapport la sortie désirée Y_p .

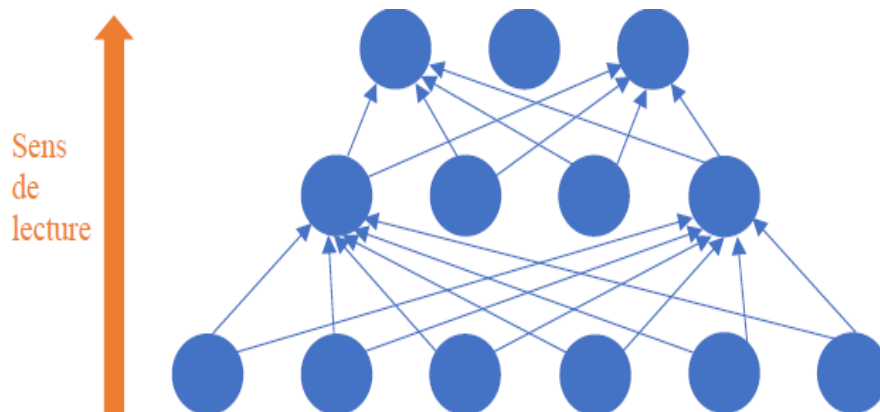


Figure 12 : la forwad-propagation

- La Backward-propagation: l'erreur calculée se propage en arrière pour calculer l'erreur individuelle pour chaque neurone, et effectuer des changements appropriés des poids du réseau de neurones afin de minimiser l'erreur.

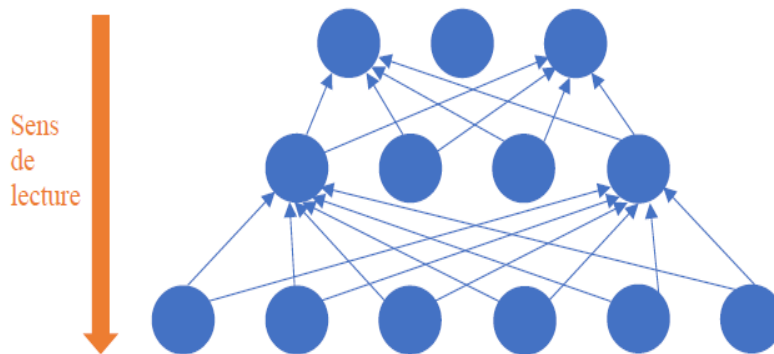


Figure 13 : La Backward-propagation

8 Topologies d'un réseau de neurones artificiels et apprentissage profond :

L'**apprentissage profond** (en anglais *Deep Learning*) est un apprentissage réalisé sur un réseau de neurones avec plusieurs couches cachées. Son principe repose sur un apprentissage hiérarchique couche par couche. Entre chaque couche interviennent des transformations non linéaires et chaque couche reçoit en entrée la sortie de la couche précédente. L'approche d'apprentissage utilisée doit donc prendre en compte les couches de neurones cachées (ex : la rétro-propagation).

Parmi les topologies de réseaux de neurones les plus utilisées dans ce sens, nous citons les suivantes.

8.1 Les réseaux de neurones convolutifs

Un réseau de neurones convolutif (CNN de l'anglais *Convolutional Neural Network*) est une classe de réseaux neuronaux artificiels dont l'architecture des connexions est inspirée de celle du cortex visuel des mammifères.

Les réseaux de neurones à convolution sont capables de catégoriser les informations des plus simples aux plus complexes. Ils consistent en un empilage de plusieurs couches de neurones, des fonctions mathématiques à plusieurs paramètres ajustables, qui prétraitent de petites quantités d'informations. Les CNNs sont caractérisés par leurs quatre types de couches pour un réseau de neurones convolutif : la couche de convolution, la couche de pooling, la couche de correction ReLU et la couche entièrement connectée (ou *fully-connected, FC*).

- La couche de convolution (CONV) qui traite les données d'un champ récepteur.
- La couche de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage).
- La couche de correction (ReLU), souvent appelée par abus ReLU en référence à la fonction d'activation (*Rectified Linear Units* présentée précédemment).

- La couche entièrement connectée (FC), qui est une couche de type perceptron.
- La couche de perte (LOSS).

Les CNN sont en général les modèles les plus performants pour la classification d'images et l'analyse d'images visuelles.

8.2 Les réseaux de neurones récurrents

Dans un réseau récurrent, appelé aussi réseau bouclé ou dynamique, il est possible de trouver des connections de neurones formant un cycle où l'information peut revenir à son point de départ. Nous avons déjà présenté ce type de réseaux. Vu leur nature, ils rentrent dans le cadre de l'apprentissage profond. Ils se démarquent d'autres réseaux utilisés pour l'apprentissage profond du fait que la sortie d'un neurone puisse être fonction elle-même. Or, cela n'est possible que si la notion de temps est prise en compte. Ce type de réseaux est notamment utilisé pour la modélisation dynamique de processus non linéaires.

Caractéristiques fonctionnelles	Type de réseaux de neurones
Reconnaissance de formes	MPL, Hopfield, Kohonen,
Mémoire associatives	Hopfield, MLP récurrents, Kohonen
Optimisation	Hopfield, CNN
Approximation de fonctions	MLP
Modélisation et control	MLP, MLP récurrent,
Traitement d'image	CNN, Hopfield
Classification et clustering	MLP, Kohonen, RBF

Tableau 1 : Quelques caractéristiques fonctionnelles et leurs types de RNA.

9 Défis relatifs à l'utilisation des réseaux de neurones

Les RNA sont considérés comme une nouvelle approche de traitement de l'information par apprentissage de cette information et la rendent disponible à l'utilisation afin de résoudre un tel problème.

Il existe de nombreux problèmes qui peuvent être résolus par un réseau de neurones. Toutefois, les réseaux de neurones sont généralement utilisés pour traiter des types particuliers de problèmes. Les quatre types de problèmes qui sont souvent résolus par les réseaux de neurones sont les suivants [39] :

9.1 La classification

La classification est le processus de classement des entrées en groupes. Par exemple, une compagnie d'assurance peut vouloir classer les demandes d'assurance dans les différentes catégories de risque, ou une organisation en ligne peut vouloir de son système de messagerie de classer le courrier entrant dans des groupes de messages spam et non-spam ainsi la détection des Fake News.

9.2 La prédiction

La prédiction est une autre application des réseaux neuraux. Etant donné une série temporelle de données d'entrée, un réseau de neurones peut prédire les valeurs futures. La précision de la prédiction dépend de nombreux facteurs, tels que la quantité et la pertinence des données d'entrée. Par exemple, les réseaux de neurones sont généralement appliqués à des problèmes de prédiction de l'évolution des marchés financiers.

9.3 La reconnaissance des formes

Reconnaissance des formes est l'une des utilisations les plus courantes des réseaux neuraux. Pattern recognition est une forme de classification et est tout simplement la capacité de reconnaître un motif. Le modèle doit être reconnu même s'il n'est pas clair. Exemple : la reconnaissance des visages.

9.4 L'optimisation

Une autre application des réseaux de neurones est l'optimisation [11] qui peut être appliquée à de nombreux problèmes pour lesquels une solution est recherchée. Le réseau de neurones peut ne pas toujours trouver la solution optimale mais il cherche à trouver une solution acceptable.

Les RNA sont utilisés dans le domaine des sciences cognitives où nous cherchons à développer des modèles capables de manifester des capacités d'apprentissage et d'adaptation à leur environnement. Les deux concepts de base à l'origine des différents types de modèles neuronaux sont l'architecture et l'apprentissage.

10 Conclusion

Dans ce chapitre, nous avons expliqué la signification de l'apprentissage automatique d'une manière générale. Nous avons vu les différents types d'apprentissage ainsi que quelques algorithmes utilisés pour l'apprentissage. Nous avons aussi présenté différentes architectures et topologies des réseaux de neurones et les défis relatifs à leur utilisation.

L'idée à la base du fonctionnement des réseaux de neurones est de s'inspirer des propriétés du cerveau pour construire des systèmes de calcul capables d'effectuer des opérations de classification par apprentissage. Leur principal avantage par rapport aux autres modèles est leur capacité d'apprentissage et de généralisation de leurs connaissances à des entrées inconnues.

A cet effet, le chapitre suivant sera consacré à la présentation du principe de classification des textes avec des réseaux de neurones.

Chapitre 2: Classification de textes

1 Introduction

La classification de textes (ou catégorisation de textes) est une tâche qui consiste à assigner une ou plusieurs classes, parmi une liste prédéfinie ou non de classes, à un document. C'est actuellement un domaine de recherche très actif et bien établi. L'automatisation de cette tâche est devenue un enjeu de taille pour la communauté scientifique.

Le problème de détection des Fake News est abordé dans ce mémoire comme un problème de catégorisation de textes à deux catégories (ou classes) : la catégorie des Fake News pour les fausses informations, et la catégorie Good News pour les vraies informations. Il est donc important de bien comprendre ce qu'est la classification de textes avant d'attaquer le problème de détection de Fake News.

Dans ce chapitre, nous présentons les concepts de bases relatifs à la catégorisation de textes ainsi que le processus de classification de texte qui est constitué de plusieurs étapes : prétraitement de textes, la représentation des textes et types de classifieurs utilisés. Nous exposons par la suite les applications de la classification de textes. Nous citons les principales difficultés liées à la catégorisation, enfin nous terminons par une conclusion.

2 Définition

La classification de textes [12] (appelée aussi catégorisation de textes) est une tâche originalement associée à la recherche d'information. Son objectif est de classer de manière automatisée des documents en se basant sur le contenu de texte et certaines métadonnées (auteur, la date de production, etc.).

Elle connaît ces derniers temps un fort regain d'intérêt. Cela est dû à l'augmentation de la disponibilité des documents sous forme numérique et de la nécessité de les organiser. C'est une discipline qui a connu des progrès considérables à partir des années 90 avec l'apparition d'algorithmes d'apprentissage automatique.

Pour, la recherche en catégorisation de textes a été encouragée par l'entrée, ces dernières années, de la communauté d'apprentissage automatique dans ce domaine. L'avantage de classification de textes pour ces méthodes est qu'il existe souvent un corpus d'apprentissage contenant des textes déjà classés. En effet, la tâche de classification de textes s'inscrit dans un processus d'automatisation d'une tâche récemment effectués manuellement. C'est typiquement le cas où un expert a déjà élaboré les catégories, a indexé et classé quelques documents. Le besoin d'automatisation de cette tâche est accentué avec la surabondance des textes entrants. [13]

Nous pouvons distinguer trois types de classifications de textes :

- Classification binaire (bi-classe) : ce type de classification correspond au filtrage, elle permet de répondre aux questions suivants par exemple «La News est Fake ou pas ?», «Le courriel est un spam ou non ?».
- Classification multi-classes disjointes : c'est une classification à n classes mais le document doit être affecté à une et une seule classe. Un système de classification multi-classes disjointes répond à la question « À quelle classe appartient ce document ? »
- Classification multi-classes : c'est une classification où nous pouvons associer un texte à une ou plusieurs classes ou bien aucune classe. Le système de classification répond à la question « A quelles classes appartient ce document ? », et ce type de classification c'est le cas le plus général de la classification.

3 Processus de la catégorisation de textes

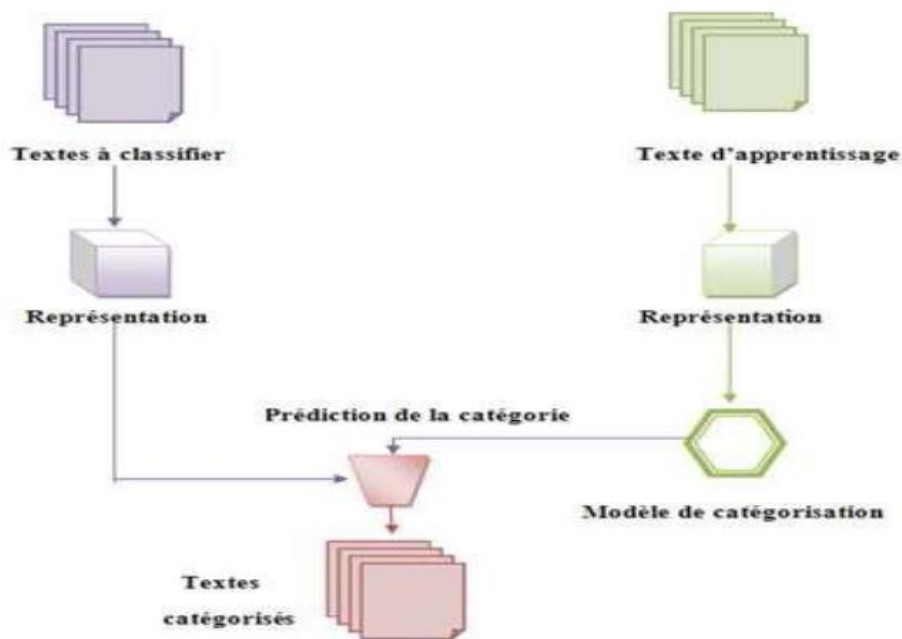


Figure 14 : Processus de catégorisation.

Comme il vient d'être mentionné, le but de la catégorisation automatique des textes est d'apprendre à une machine à classer un texte dans la bonne catégorie en se basant sur son contenu. Pour identifier la catégorie d'un texte, ou la classe à laquelle un texte est associé, un ensemble d'étapes est habituellement suivi.

3.1 Prétraitement du texte

Pour entraîner et créer un classifieur à partir d'un corpus de texte, nous devons impérativement transformer les textes en entrées valides compréhensibles par différents algorithmes de classification ou de clustering. Ces entrées sont en fait des vecteurs ou des matrices qui définissent le poids de chaque descripteur (mot ou groupe de mots) dans chaque texte où ils apparaissent, cette transformation entre le document texte et document vecteur est appelée prétraitement.

3.1.1 Lemmatisation et Racinisation

Dans certaines représentations vectorielles, chaque mot présent dans le corpus est considéré comme un descripteur. Cette façon de faire contient certaines irrégularités notamment par les verbes à l'infinitif et les verbes conjugués (exemple : transformer, transforme, transformons, etc.) ces mots ont les mêmes sens mais chacun est considéré comme un descripteur à part entière.

La racinisation vient résoudre ce problème en considérant uniquement la racine de ces mots plutôt que les mots en entier sans se soucier de l'analyse grammaticale.

La lemmatisation [14] est une forme compliquée de la racinisation puisqu'elle se base sur une analyse grammaticale pour ramener les termes à leur forme canonique. En particulier, elle met tous les mots au singulier, les adjectifs au masculin singulier, et tous les verbes conjugués à l'infinitif.

La substitution des mots par leurs racines permet une meilleure représentation, surtout dans le cas de la lemmatisation, et réduit considérablement la dimension des descripteurs.

Les confusions et les ambiguïtés peuvent avoir lieu en représentant par le même descripteur des mots avec des sens différents. Comme par exemple « glace » qui peut être un miroir ou « une glace aux chocolats ». L'ambiguïté peut être causée aussi par le sens de remplacement de la forme pluriel d'un mot par sa forme singulier comme « actions » qui est représenté par le descripteur « action » dans un texte économique, le mot « actions » qui signifie les actions des entreprises n'a rien avoir avec la notion d'action employée par exemple dans cette phrase « un film d'action ».

3.1.2 Mots vides

Après le découpage du texte en tokens, nous apercevons que les articles, les prépositions, les déterminants, les adverbes (comme la, le, les, dans) dans la langue française représentent 30% des mots dans le texte du corpus, c'est ce que nous appelons « le mot vide ». La présence de ces mots n'apporte souvent aucune différence sur le plan sémantique ou lexical. Cela veut dire que ces termes peuvent être écartés du corpus pour en réduire la dimension.

Ces mots doivent être supprimés de la représentation des textes pour deux raisons :

- D'un point de vue linguistique, ces mots ne comportent que très peu d'informations. La présence ou l'absence de ces mots n'aident pas à deviner le sens d'un texte. Pour cette raison, ils sont communément appelés « mots vides ».
- D'un point de vue statistique, ces mots se retrouvent sur l'ensemble des textes sans aucune discrimination et ne sont d'aucune aide pour la classification. [14]

3.1.3 Nettoyage des mots

La segmentation consiste à découper la séquence des caractères afin de regrouper les caractères formant un même mot. Cette étape permet d'isoler les ponctuations (reconnaissance des fins de phrase ou de paragraphe), ensuite découper les séquences de caractères en fonction de la présence ou l'absence de caractères de séparation (de type « espace », « tabulation » ou « retour à la ligne »), puis regrouper les chiffres pour former des nombres (reconnaissance éventuelle des dates), de reconnaître les mots composés. Eventuellement, nous pouvons unifier les écritures en lettres majuscules ou en lettres minuscules avant ou après les opérations déjà indiquées. C'est un traitement de surface assez simple dans le principe, mais particulièrement difficile à réaliser de manière exacte sur les documents ayant beaucoup de bruits et des représentations assez variées. Notons que pour des corpus multilingues, une technique de segmentation moins intuitive a été proposée : la segmentation en n-grammes.

Malgré ces premières transformations, la taille de vocabulaire reste un véritable problème que nous devons résoudre car si tous les mots présents dans les documents de l'espace d'apprentissage sont utilisés, nous nous retrouvons face à un espace vectoriel ayant une dimension très large. Le traitement d'un tel espace nécessite beaucoup de mémoire et de temps de calcul. Pour résoudre ce problème des techniques de réduction de la taille du vocabulaire sont utilisées. Cette réduction permet de réduire l'espace de stockage nécessaire pour les données textuelles, ainsi que le temps d'apprentissage et d'exploitation des algorithmes de décision.

La réduction de dimension est l'étape primordiale dans le processus de prétraitement des textes qui a pour objectif de sélectionner ou d'extraire un sous-ensemble optimal de caractéristiques pertinentes pour un critère fixé. Ces techniques de réduction sont classées comme suit :

Sélection d'attributs : La réduction de la taille de l'espace d'apprentissage par la méthode de sélection d'attributs vise à diminuer la taille de l'espace d'apprentissage de $|T|$ à une taille $|T'| < |T|$ en sélectionnant uniquement un sous-ensemble des attributs existants.

Extraction d'attributs : Contrairement aux techniques de sélection d'attributs qui visent à proposer par sélection un sous ensemble des attributs existants, l'extraction des attributs a, par définition, pour objectif de proposer, via une synthétisation, un sous ensemble $|T'| < |T|$ composé de nouveaux attributs à partir des attributs existants. Ce processus consiste à créer à partir des attributs originaux un sous ensemble d'attributs synthétiques qui maximise l'efficacité de la classification et qui élimine les problèmes liés aux synonymies, homonymies et polysémie.

3.2 La représentation des textes

Quelle que soit la méthode de classification retenue, la première opération consiste à représenter les documents de façon à ce qu'ils puissent être traités automatiquement par les classifieurs.

Les textes en langage naturel ne peuvent pas être directement interprétés par un classifieur ou par les algorithmes de classification (le classifieur est incapable de traiter directement des données structurées comme les textes ou non structurées comme les images, son, les vidéos, etc.) pour cela une technique de représentation numérique est nécessaire.

La représentation de textes et la phase la plus importante dans le processus de classification, son rôle est d'interpréter les textes et les représenter sous forme exploitable par la machine.

Pour la représentation des documents textuels, plusieurs méthodes sont utilisées, nous présentons quelques-unes dans ce qui suit.

3.2.1 Représentation en sac de mots (*Bag of Words*)

C'est la représentation la plus couramment utilisée [17] dans laquelle les textes sont transformés simplement en vecteurs dont chaque composante représente un mot. Les composants de vecteur peuvent être une fonction d'occurrence des mots dans le texte. Cette représentation exclut toute analyse grammaticale et toute notion de distance entre les mots, et c'est pourquoi elle est appelée « sac de mot ». Avec cette approche, les documents sont représentés par des vecteurs de dimension égale à la taille du vocabulaire, qui est en générale assez grand. Par exemple :

<p>Marionnaud: Union et Etudes Investissement franchit 5% des droits de vote PARIS, 31 juil (AFP) - La société Union et Etudes Investissement (caisse Nationale de Crédit Agricole) a franchi en hausse le seuil de 5% des droits de vote du groupement français de parfumerie Marionnaud et détient désormais 292.157 actions, soit 8,09% du capital et 5,05% des droits de vote, a indiqué vendredi le Conseil des Marchés Financiers. Ce franchissement de seuil résulte de l'acquisition de 11.460 actions, précise le CMF.</p>					
a	2	détient	1	le	3
acquisition	1	en	1	marchés	1
actions	2	et	4	marionnaud	2
agricole	1	études	2	nationale	1
caisse	1	financiers	1	parfumerie	1
capital	1	franchi	1	précise	1
ce	1	franchissement	1	résulte	1
cmf	1	franchit	1	seuil	2
conseil	1	français	1	société	1
crédit	1	groupement	1	soit	1
de	9	hausse	1	union	2
des	4	indiqué	1	vendredi	1
droits	3	investissement	2	vote	3
du	2	l	1		
désormais	1	la	1		

Figure 15 : Exemple d'un texte et de son vecteur associé. Les composantes du vecteur sont Simplement les occurrences des mots du texte.

Mais présente plusieurs inconvénients qui sont les suivants :

- Les mots composés allemands peuvent être très complexes, ex : Levensversicherungsgesellschaftsangestellter (employé d'une société d'assurance vie). Même en français nous trouvons des mots composés comme arc-en-ciel, peut-être, etc.
- Le chinois et le japonais ne séparent pas les mots par des espaces, ce qui peut mener à plusieurs segmentations.
- L'arabe et l'hébreu sont écrits de droite à gauche, mais certains éléments tels que les nombres sont écrits de gauche à droite.

3.2.2 Représentation par phrases

Un certain nombre de chercheurs [13] [17] proposent d'utiliser les phrases comme unité de représentation à la place des mots à cause des problèmes syntaxiques dans la représentation précédente en « sac de mots », puisque les phrases sont plus informatives que les mots seuls. Par exemple « recherche d'information », « world wide web », ont un degré plus petit d'ambiguïté que les mots constitutifs. Aussi, les phrases ont l'avantage de conserver l'information relative à la position du mot dans la phrase.

3.2.3 Représentation avec les racines lexicales

Cette méthode consiste à remplacer les mots du document par leurs racines lexicales, et à regrouper les mots de la même racine dans une seule composante [18]. Ainsi, plusieurs mots du document seront remplacés par la même racine, cette méthode peut être réalisée en utilisant un des algorithmes les plus connus pour la langue anglaise qui est l'algorithme de Porter de normalisation de mots qui sert à supprimer les affixes de ces derniers pour obtenir une forme canonique. Néanmoins, la transformation automatique d'un mot vers sa racine lexicale peut engendrer certaines anomalies. En effet, une racine peut être commune pour des mots qui portent des sens différents tel que les mots jour, journalier, journée ont la même racine « jour » mais se rendent à trois notions différentes, cette représentation dépend aussi de la langue utilisée.

3.2.4 Représentation avec les lemmes

La lemmatisation [19] consiste à utiliser l'analyse grammaticale afin de remplacer les verbes par leur forme infinitive et les noms par leur forme au singulier. En effet, Un mot donné peut avoir différentes formes dans un texte, mais leur sens reste le même. Par exemple, les mots jouons, joueurs, jouet seront remplacés par leurs lemmes : « jouer », « joueur » et « jouet » selon le contexte. Cette représentation est simple mais elle peut causer une perte d'informations donnée par le contexte nécessaire à la distinction des lemmes polysémiques (possèdent plusieurs sens) et la présence de synonymes, considérés comme des lemmes différents même s'ils font référence au même concept.

3.2.5 Représentation avec les n-grammes

Cette méthode consiste à représenter le document par des n-grammes [20]. Le n-gramme est une séquence de n caractères consécutifs. Elle consiste à découper le texte en plusieurs séquences de n caractères en se déplaçant avec une fenêtre d'un caractère.

Cette technique présente plusieurs avantages. Les n-grammes capturent automatiquement les racines des mots les plus fréquents sans passer par l'étape de recherche des racines lexicales, de plus c'est une méthode indépendante de la langue.

3.2.6 Représentation conceptuelle

La représentation conceptuelle [21] sert à représenter le document sous forme d'un ensemble de concepts, ces derniers peuvent être capturés en utilisant les réseaux sémantiques

Cette méthode a comme avantage de réduire l'espace de représentation car les mots qui sont synonymes partagent le même concept. Cependant, l'inconvénient majeur de cette représentation est qu'il n'existe pas des bases lexicales pour toutes les langues.

3.2.7 Représentation dans les réseaux de neurones

Une approche fondée sur les réseaux de neurones a été proposée dans la thèse de Wiener dont les résultats ont été repris dans [22]. Deux architectures neuronales sont proposées et testées sur le corpus Reuters¹-22173 (qui est une ancienne version du corpus Reuters-21578 disponible aujourd'hui).

La première architecture est un perceptron multicouche avec une couche de neurones cachés et un neurone de sortie (cette architecture est présentée au premier chapitre); un réseau de neurones différent est construit pour chaque catégorie. Les descripteurs sont sélectionnés soit par une méthode de sélection de termes.

Pour la deuxième architecture, les catégories du corpus Reuters sont regroupées en cinq grands ensembles (*agriculture, energy, foreignexchange, government, metals*). Un réseau est ensuite utilisé pour déterminer à quel ensemble appartient un document, puis cinq réseaux différents sont construits pour déterminer, à l'intérieur d'un ensemble, la catégorie exacte du document. Cette architecture a l'avantage de permettre à chacun des cinq réseaux d'être "spécialisé" et d'utiliser une représentation particulièrement adaptée pour distinguer des catégories proches. Cette deuxième architecture améliore les résultats, mais elle nécessite un découpage manuel des catégories pour déterminer les ensembles et n'est réalisable que sur un corpus pour lequel le nombre de catégories est connu à l'avance et n'évolue pas.

4 Types de classifieurs utilisés

La classification de textes comporte un choix de technique d'apprentissage(ou classifieurs) disponibles. Parmi les méthodes d'apprentissage les plus souvent utilisées figurent les machines à vecteurs de support, naïve bayes et réseaux de neurones.

4.1 Les machines à vecteurs de support

L'algorithme SVM (Support Vector Machine) est une méthode d'apprentissage supervisé introduite pour résoudre un problème de la reconnaissance de l'écriture en passant par la catégorisation de textes [17].

La méthode SVM est une classification linéaire utilisant des mesures de distance. En ce qui concerne son application à la problématique de catégorisation de texte, l'algorithme repose sur une interprétation géométrique simple. L'idée générale est de représenter dans un espace vectoriel

¹ Le corpus Reuters-21578 est un ensemble de dépêches financières .Ce corpus est une mise à jour du corpus Reuters-22173. Cette mise à jour, effectuée en 1996, a permis de supprimer les documents présents deux fois, de corriger des erreurs typographiques, de préciser certains formats, et de mieux définir le découpage à considérer pour l'apprentissage et le test..

multidimensionnel des textes, où chaque texte est un point dans cet espace et de rechercher une règle de décision pour les nouveaux textes.

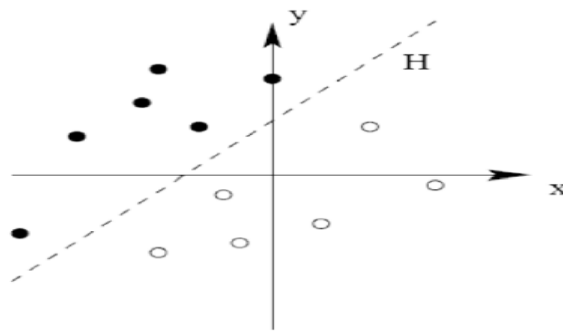


Figure 16: la séparation du l'hyper plan par les SVM.

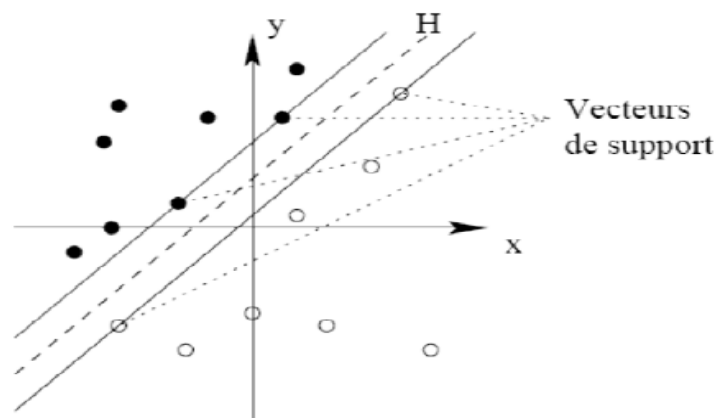


Figure 17: Les vecteurs de support.

Ainsi l'approche par SVM permet de définir par apprentissage une surface de séparations entre des exemples minimisant le risque d'erreurs et maximisant la marge entre deux classes, elle s'appartient aux classifieurs linéaires, dans le sens où elles tentent de séparer l'espace en deux mais certaine manipulation mathématique les rendes adaptables à des problèmes non linéaires.

4.2 Les classifieurs naïfs de Bayes

Naïve-Bayes est un algorithme d'apprentissage supervisé [21] utilisé notamment pour la classification de textes. Ce modèle est le représentant le plus populaire des classifieurs probabilistes. Il est basé sur le théorème de Bayes.

Le modèle Naïve-Bayes vise à estimer la probabilité conditionnelle qu'un document appartienne à une catégorie donnée en sachant son contenu. La partie Naïve de ce modèle est l'hypothèse d'indépendance des mots. C'est à dire que la probabilité conditionnelle relative à l'apparition d'un mot sachant une catégorie est supposée être indépendante de l'apparition des autres mots.

4.3 Les classifieurs à base de réseaux de neurones

Les réseaux de neurones artificiels [22] sont habituellement utilisés pour des tâches de classification. Le connexionnisme peut être défini comme le calcul distribué d'unités simples, regroupées en réseau. Un réseau de neurones est un ensemble d'éléments ou unités extrêmement simples (neurones) se comportant comme des fonctions de seuil, suivant une certaine architecture. Chaque neurone prend en entrée une combinaison des signaux de sortie de plusieurs autres neurones, affectés de coefficients (les poids)

Un réseau de neurones en classification de texte est un réseau d'unités construit à partir des documents d'apprentissage où les unités d'entrée représentent les pondérations des termes, les unités de sortie représentent les catégories.

Plusieurs travaux, dans le domaine de classification automatique de textes, à base des approches neuronales ont été évoqués. Une approche fondée sur les réseaux de neurones a été proposée dans la thèse de Wiener dont les résultats ont été repris dans [22]. Deux architectures neuronales sont proposées et testées sur le corpus Reuters.

5 Les applications de la classification des textes

Aujourd'hui, la classification de textes est appliquée dans une variété de contextes [13], il s'agit d'un support pour différentes applications parmi lesquelles :

- Le filtrage de documents.
- L'indexation des documents.
- La réponse à des requêtes sur le web.
- Routage des courriels.
- Détection des intrusions.

6 Défis de la catégorisation de textes

Plusieurs difficultés peuvent s'opposer au processus de catégorisation de textes, les principales sont les suivantes :

La redondance : La redondance et la synonymie permettent d'exprimer le même concept par des expressions différentes. Il y a ainsi plusieurs façons d'exprimer la même chose. Cette difficulté est liée à la nature des documents traités qui sont exprimés en langage naturel contrairement aux données numériques. Le-Fevre illustre cette difficulté dans l'exemple du chat et l'oiseau : *mon chat mange un oiseau, mon gros matou croque un piaf et mon félin préféré dévorer une petite bête à plumes*. La même idée est représentée de trois manières différentes, différents termes sont utilisés d'une expression à une autre mais en fin compte c'est bien le malheureux oiseau qui est dévoré par ce chat.

L'ambiguïté : À la différence des données numériques, les données textuelles sont sémantiquement riches, du fait qu'elles sont conçues et raisonnées par la pensée humaine. À cause de l'ambiguïté, les mots sont parfois de mauvais descripteurs ; par exemple le mot glace prend deux significations.

La graphie : Un terme peut comporter des fautes d'orthographe ou de frappe comme il peut s'écrire de plusieurs manières ou s'écrire avec une majuscule. Ce qui va peser sur la qualité des résultats. Parce que si un terme est orthographié de deux manières dans le même document, la simple recherche de ce terme avec une seule forme graphique néglige la présence du même terme sous d'autres graphies.

Complexité de l'algorithme d'apprentissage : Un texte est représenté généralement sous forme d'un vecteur contenant les nombres d'apparitions des termes dans ce texte. Or, le nombre de textes traités est très important sans oublier le nombre de termes composant le même texte. Ainsi, la dimension du tableau (textes * termes) à traiter complique considérablement la tâche de classification en diminuant la performance du système.

Présence-Absence de termes : La présence d'un mot dans le texte indique un propos que l'auteur a voulu exprimer. Ainsi, il existe une relation d'implication entre le mot et le concept associé. Il y a aussi plusieurs façons d'exprimer les mêmes choses (peut-être même avec plusieurs mots). Dès lors l'absence d'un mot n'implique pas obligatoirement que le concept qui lui est associé est absent du document. Cette réflexion pointue nous amène à être attentifs quant à l'utilisation des techniques d'apprentissage se basant sur l'exclusion d'un mot particulier.

Les mots composés : La non-prise en charge des mots composés (comme : *Arc-en-ciel, peut-être, sauve-qui-peut*, etc.) peut poser problème. En effet, leur nombre est très important dans toutes les langues, et traiter le mot *Arc-en-ciel* par exemple en étant 3 termes séparés réduit considérablement la performance d'un système de classification. Néanmoins, l'utilisation de la technique des n-grammes pour le codage des textes atténue considérablement ce problème des mots composés.

7 Conclusion

Dans ce chapitre, nous avons donné des notions sur la classification automatique de données textuelles, nous avons commencé par une définition sur la catégorisation des textes puis nous avons décrit le processus de la classification (l'importance de chaque étape de prétraitement et les différentes techniques de représentation de texte). Nous avons également introduit les différents choix possibles d'un classifieur.

La catégorisation de textes a essentiellement progressé ces dix dernières années grâce à l'introduction des techniques héritées de l'apprentissage automatique qui ont amélioré très significativement les taux de bonne classification.

Le chapitre suivant présente les différentes notions des Fake News ainsi que le modèle adopté pour leur représentation.

Chapitre 3: Les Fake News

1 Introduction

Les Fake News prennent aujourd'hui une place de plus en plus importante sur le web. Ce phénomène de diffusion de fausses informations et de rumeurs vise la plupart du temps à nuire à des personnes, des organisations ou même à des processus démocratiques. En réaction, les plus grands acteurs du web ont mis en place des moyens importants pour contrôler la diffusion des Fake News et tenter d'endiguer ce phénomène. Des méthodes de détection automatiques fondées sur l'analyse de texte ont ainsi été développées pour les détecter.

Les Fake News existent depuis des décennies et ce n'est pas un nouveau phénomène. Ce qui l'est en revanche, c'est son ampleur et la vitesse avec laquelle les Fake News se répandent, grâce aux médias numériques. Bien que la quantité de Fake News de nos jours semble continuer à augmenter, certains peuvent trouver de plus en plus difficile de déchiffrer ce qui est vrai et ce qui ne l'est pas.

Dans ce chapitre, nous définissons tout d'abord le concept de Fake News puis nous intéressons aux objectifs de diffusion des Fake News. Ensuite, nous abordons la problématique de la représentation des Fake News, et nous présentons en particulier le modèle de représentation de textes NNLM. Enfin, et avant de conclure, nous présentons la topologie choisie pour notre réseau de neurones qui nous servira pour classifier les News.

2 Définition

Depuis que Donald Trump a été élu, cette expression « Fake News ! » est partout et le président des Etats-Unis fait partie des premiers à s'en servir.

Fake News (ou fausses nouvelles) sont des fausses informations qui ont l'air sérieuses et authentiques, parfois publiées avec un style journalistique qui copie les médias traditionnels. Mais à l'inverse d'articles publiés sur des sites respectés et reconnus comme CNN, BBC, ces Fake News n'offrent aucune source pour appuyer leurs propos. Elles ont souvent des titres à sensation, contiennent des propos ou images parfois choquantes, souvent au sujet de figures publiques ou d'événements historiques dans le but de répandre des fausses rumeurs ou de fausses théories de complot.

Littéralement, ce sont des nouvelles truquées, fabriquées et diffusées volontairement. Elles ressemblent souvent à de vraies informations. Certains s'y laissent prendre et les partagent sur le Web et les réseaux sociaux, contribuant ainsi à leur propagation. [24]

3 Objectifs de leur publication

Ces dernières années, le phénomène des Fake News s'étend sur le web aux dépens des internautes. Ces fausses informations peuvent être propagées dans des buts différents. Nous soulevons quelques-uns de ces objectifs dans ce qui suit.

Les Fake News peuvent être propagées en vue de tromper les gens et sont ancrées dans des intentions espiègles : servir un intérêt idéologique ou politique, faire de l'argent, etc. Par exemple, une personne ou une population pourrait fonder une décision importante (concernant sa santé, ses choix politiques, ses déplacements, sa carrière, etc.) en se basant sur ces Fake News.

Il arrive aussi qu'une fausse nouvelle ait pour objectif une arnaque comme d'escroquer des gens, de convaincre de divulguer des renseignements personnels, ou d'amener à cliquer sur un lien qui installera un logiciel malveillant.

Dans une autre mesure, les Fake News peuvent être propagées à des fins de canulars ou de blagues. En effet, il est parfois difficile de faire la différence entre une fausse actualité et une forme d'humour ou de satire, même si souvent la source est connue pour ses parodies. Dans ce cas, les détails et le ton de l'article sont simplement satiriques. Des sites satiriques légitimes, comme *Le Gorafi* et le journal de *Mourréal*, sont amusants et divertissants mais certains peuvent confondre leur contenu avec de vraies nouvelles.

Les Fake News peuvent aussi être utilisées pour faire de la publicité mensongère. En effet, la plupart des sites Internet sont financés en grande partie par la publicité en ligne. Plus il y a d'audience, plus ils gagnent de l'argent. Les Fake News, si bien rédigés, font beaucoup d'audience et génèrent énormément de trafic et donc de revenus. [26]

4 Représentations des Fake News

Pour représenter les Fake News, des modèles de représentation de textes existants peuvent être utilisés. Il n'y a pas de modèles spécifiques aux Fake News. Cependant, d'autres informations en plus du texte peuvent être ajoutées à la représentation de la News. Ainsi, les métadonnées relatives à la News sont utiles et donnent plus d'information quant à la véracité de la News.

4.1 NNLM

Le modèle de langage de réseau neuronal (NNLM) [29] est un modèle de langage¹ basé sur les réseaux de neurones, exploitant leur capacité à apprendre des représentations distribuées pour réduire l'impact de la « malédiction » de la dimensionnalité .

Dans le contexte des algorithmes d'apprentissage, la « malédiction de la dimensionnalité » fait référence à la nécessité d'avoir un grand nombre d'exemples d'entraînement lors de l'apprentissage de fonctions très complexes. Lorsque le nombre de variables d'entrée augmente, le nombre d'exemples requis peut augmenter de manière exponentielle.

Une représentation distribuée d'un symbole est un tuple (ou vecteur) d'entités qui caractérisent la signification du symbole et qui ne s'excluent pas mutuellement. Si un humain choisissait les caractéristiques d'un mot, il pourrait choisir des caractéristiques grammaticales comme le genre ou la pluralité, ainsi que des caractéristiques sémantiques telles qu'animées ou invisibles. Avec un modèle de langage de réseau neuronal, l'algorithme d'apprentissage permet de découvrir ces caractéristiques. , et les fonctionnalités sont à valeur continue (ce qui rend le problème d'optimisation impliqué dans l'apprentissage beaucoup plus simple).

L'idée de base est d'apprendre à associer chaque mot du dictionnaire à une représentation vectorielle à valeur continue. Chaque mot correspond à un point dans un espace d'entités. On peut imaginer que chaque dimension de cet espace correspond à une caractéristique sémantique ou grammaticale des mots. L'espoir est que les mots fonctionnellement similaires deviennent plus proches les uns des autres dans cet espace, au moins dans certaines directions. Une séquence de mots peut ainsi être transformée en une séquence de ces vecteurs caractéristiques appris. Le réseau neuronal apprend à mapper cette séquence de vecteurs de caractéristiques à une prédiction d'intérêt, telle que la distribution de probabilité sur le mot suivant dans la séquence. Ce qui pousse les traits de mots appris à correspondre à une forme de similarité sémantique et grammaticale est que lorsque deux mots sont fonctionnellement similaires, ensemble de formation , l'ensemble des séquences de mots utilisées pour former le modèle.

L'avantage de cette approche de représentation distribuée est qu'elle permet au modèle de se généraliser aux séquences qui ne sont pas dans l'ensemble des séquences de mots d'apprentissage, mais qui présentent des caractéristiques similaires, c'est-à-dire leur représentation distribuée. Comme les réseaux neuronaux ont tendance à mapper les entrées à proximité des sorties à proximité, les prédictions correspondant à des séquences de mots ayant des

¹ Un modèle de langage est une fonction, ou un algorithme d'apprentissage, qui capture les caractéristiques statistiques saillantes de la distribution de séquences de mots dans un langage naturel

caractéristiques similaires sont associées à des prédictions similaires. Étant donné que de nombreuses combinaisons de valeurs de caractéristiques sont possibles, un très grand nombre de significations possibles peuvent être représentées de manière compacte, permettant à un modèle avec un nombre relativement petit de paramètres de s'adapter à un grand ensemble de formation.

4.2 Représentation des métadonnées

Les données et en particulier les données numériques, prennent différentes formes. Les conversations orales, les textos et les réseaux sociaux transmettent tous des données. Les métadonnées décrivent ce que sont ces données : elles fournissent des informations sur ces données. C'est assez simple. Cependant, quand on gratte un peu la surface, on trouve que la « description » de données est un exercice technique rigoureux et chargé de connotations sociopolitiques.

Les métadonnées [37] fournissent un moyen pour classer, organiser et caractériser des données ou du contenu. L'Organisation nationale de normalisation de l'information (NISO) fournit une taxonomie qui peut être appliquée à tous les types de données ou de référentiels de données, des bibliothèques jusqu'aux sites Web, pour les données textuelles et non textuelles, en format numérique ou matériel.

La NISO décrit trois types de métadonnées :

Les métadonnées descriptives incluent des informations telles que les coordonnées, le titre ou l'auteur d'une publication, un résumé d'une œuvre, les mots clés utilisés dans un ouvrage, un emplacement géographique ou encore une explication de la méthodologie. Ces données sont utiles pour découvrir, recueillir ou regrouper des ressources en fonction des caractéristiques que partagent les ressources.

Les métadonnées structurelles expliquent la composition ou l'organisation d'une ressource. Un livre numérisé, par exemple, peut être publié sous forme d'images de chaque page, dans des fichiers PDF ou HTML.

Les métadonnées administratives sont utilisées pour gérer une ressource. Les dates de création ou d'acquisition, les autorisations et les droits d'accès ou la provenance et les lignes directrices concernant la disposition telles que le maintien ou la suppression sont des exemples de droits qu'un archiviste numérique, un curateur, pourrait employer. Les métadonnées similaires seraient appropriées pour l'administrateur d'une base de données ou pour les administrateurs responsables de capturer le flux de trafic des télécommunications ou des réseaux de données, ou de leurs registres de sécurité et leurs données concernant des attaques.

5 Topologie de notre réseau de neurones

La topologie du réseau de neurones choisie pour représenter des Fake News est celle d'un réseau entièrement connecté (Fully Connected Neural Network), ou encore appelé dense (Dense Neural Network).

Dans ce type de réseaux, les neurones dans une couche donnée ont des connexions vers toutes les sorties de la couche précédente. Leurs fonctions d'activations peuvent donc être calculées avec une multiplication matricielle suivie d'un décalage de polarisation. La représentation matricielle permet de réduire la taille du modèle sans pour autant permettre de le déployer sur des appareils avec peu de mémoire.

Pour notre modèle, nous créons un réseau de neurones de type dense (entièrement connecté). Ce modèle contient alors deux couches de neurones cachées avec 400 et 100 neurones cachés respectivement. L'entrée de la première couche de neurones cachés correspond à la sortie du modèle NNLM qui nous permet donc de représenter le texte de la News. Celui-ci prend 50 000 neurones en entrée, soit plus que le nombre de mots du vocabulaire en anglais. La sortie de la dernière couche de neurones (celle avec 100 neurones cachés) est (entièrement) connectée avec deux neurones finaux qui représentent la couche de sortie. Ces deux neurones correspondent aux deux classes *Fake* et *vraie* de la News.

6 Conclusion

Les Fake News sont largement répandues sur les réseaux sociaux. Il y a donc une énorme demande pour démystifier les fausses nouvelles. Il existe de nombreuses tentatives pour détecter les Fake News, mais le travail limité consiste à utiliser les modèles de Deep Learning.

Dans ce chapitre, nous avons présenté dans un premier temps, une définition pour les Fake News ainsi que leurs objectifs. Nous avons discuté de la représentation des Fake News et aussi de la topologie du réseau de neurones que nous avons adopté.

Le chapitre suivant sera consacré à l'implémentation d'un classifieur pour les Fake News dont la topologie est celle décrite dans ce chapitre.

Chapitre 4: Implémentation

1 Introduction

Pour classifier les News, nous avons utilisé différents outils et bibliothèques. L'implémentation ainsi réalisée nous permet ensuite de tester la pertinence de nos classificateurs sur une collection de test préexistante.

Concernant les outils utilisés, nous avons opté pour Python et l'IDE PyCharm. Ces outils facilitent grandement le processus de développement. De plus, nous avons opté pour deux bibliothèques principales, à savoir Tensorflow et Tensorflow-hub. Nous présentons les outils utilisés, ainsi que ces bibliothèques dans la section suivante. Dans la section 3, nous présentons notre code ainsi que les fonctions utilisées dans l'implémentation. Enfin, dans la section 4, nous décrivons les étapes d'implémentation et les résultats obtenus pour ce qui est de la détection de Fake News sur une collection de test existante.

2 Logiciels et bibliothèques utilisés

2.1 Logiciels

2.1.1 PyCharm

PyCharm [2] est un environnement de développement intégré (IDE) utilisé dans la programmation informatique, en particulier pour le langage python. Il est développé par la société Tchèque JetBrains. Il fournit une analyse de code, un débogueur graphique, un testeur d'unité intégré, une intégration avec des systèmes de contrôle de version et supporte le développement web avec Django.

PyCharm est multiplateforme, avec les versions Windows, Mac Os et Linux. L'édition communautaire est publiée sous la licence Apache, et il existe également une édition professionnelle publiée sous une licence propriétaire avec des fonctionnalités supplémentaires.

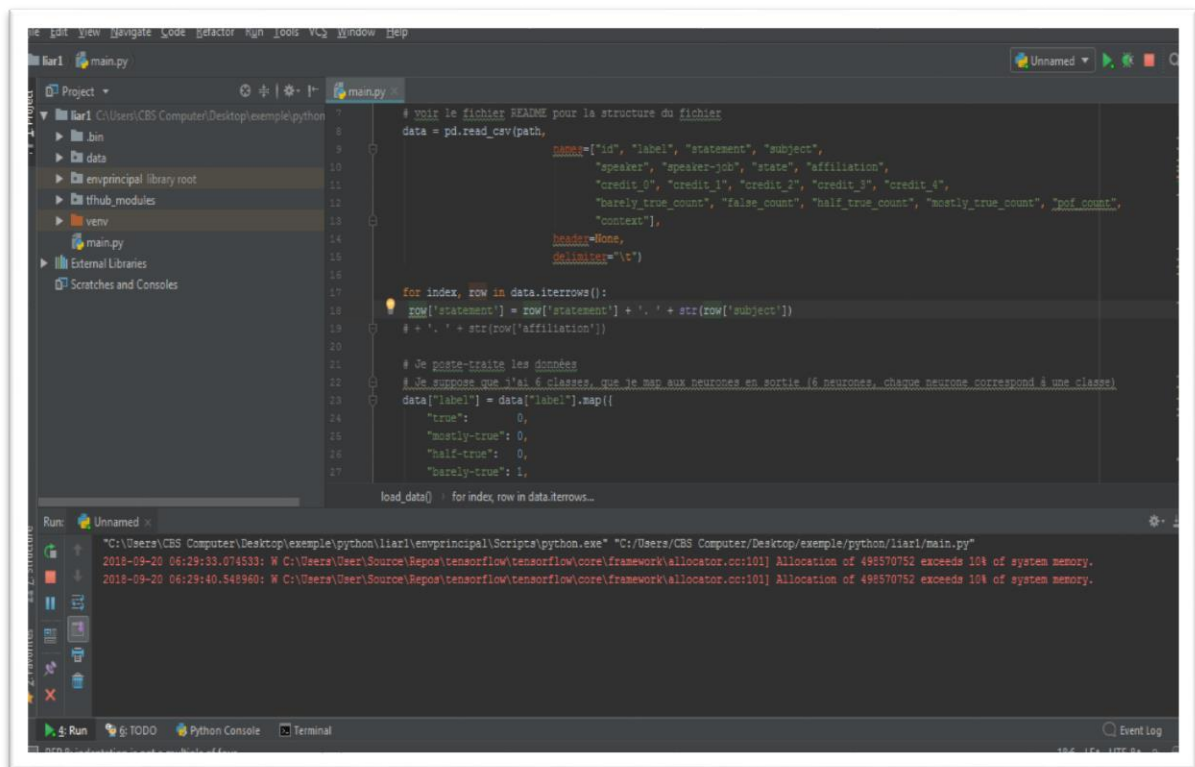


Figure 18 : Environnement PyCharm

2.1.2 ConEmu :

ConEmu (abréviation d'émulateur de console) est un émulateur de terminal à onglets gratuit et open source pour Windows. ConEmu présente plusieurs consoles et applications graphique simples sous la forme d'une fenêtre graphique personnalisable avec des onglets et une barre d'état.

2.2 Librairies

Nous avons utilisé les libraires (bibliothèques logicielles) suivantes.

2.2.1 Tensorflow

Tensorflow [30] est le second Framework d'apprentissage automatique que Google a créé et utilisé pour concevoir, construire et former des modèles d'apprentissage en profondeur.

Tensorflow est une bibliothèque open source largement utilisés dans le domaine de l'apprentissage automatique et de l'apprentissage en profondeur. La bibliothèque est populaire pour son utilisation de graphes de flux de données pour effectuer des calculs numérique.

Le nom «Tensorflow» est dérivé des opérations que les réseaux de neurones effectuant sur des tableaux de données multidimensionnels ou des tenseurs. Aujourd'hui, les principaux produits de Google utilisent Tensorflow: Gmail, Google Photos, reconnaissance de voix.

2.2.2 Tensorflow-Hub [30] [35]

Tensorflow-Hub [30] [35] est une bibliothèque pour les modules d'apprentissage automatique réutilisables dans Tensorflow. C'est une plate-forme permettant de publier, découvrir et réutiliser des parties de modules d'apprentissage automatique dans Tensorflow. Par module, nous entendons une partie autonome d'un graphique Tensorflow, ainsi que ses poids et ses ressources, qui peuvent être réutilisés dans différentes tâches dans un processus appelé apprentissage par transfert.

Tensorflow-Hub propose une variété de modules de textes à explorer notamment des modèles de langages de réseau neural dans différentes langues (EN, JP, DE et ES) ainsi que word2vec formé par Wikipédia et des versions NNLM formé sur Google actualités.

2.2.3 Pandas

Pandas [31] est une bibliothèque écrite pour python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et des séries temporelles.

Pandas est une bibliothèque open source, sous la licence BSD, fournissant des structures de données et des outils d'analyse de données hautes performances et facile à utiliser pour le langage de programmation python.

2.2.4 Numpy

Numpy [33] est une librairie python destinée à manipuler des matrices ou tableaux multidimensionnelles ainsi que des fonctions mathématiques opérant sur les tableaux.

Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau des fichiers, et manipuler des vecteurs, matrices et polynômes.

Numpy est la base de SciPy, regroupement de bibliothèques python autour du calcul scientifique.

2.2.5 Tensorboard

TensorBoard [34] est utilisé pour visualiser votre graphique Tensorflow, tracer des métriques quantitatives sur l'exécution de votre graphique et afficher des données supplémentaires telles que des images qui le traversent.

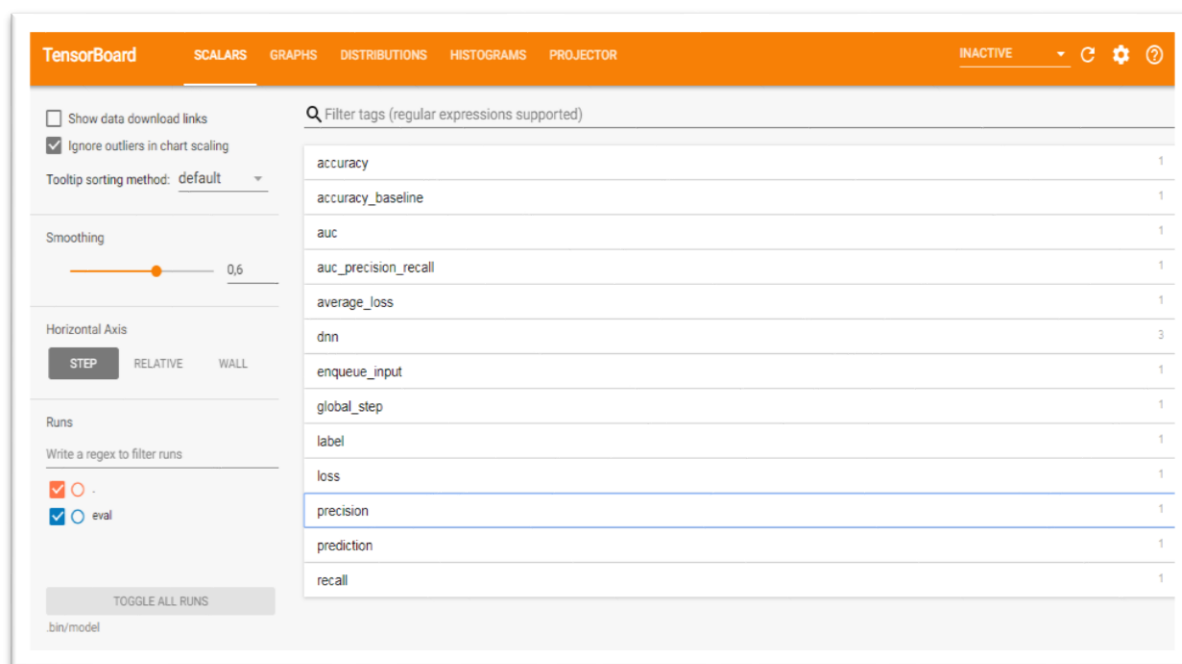


Figure 19 : Tensorboard

2.2.6 Python

Python [2] est un langage de programmation de haut niveau interprété (il n'y a pas d'étape de compilation) et orienté objet avec une sémantique dynamique. Il est très sollicité par une large communauté de développeurs et de programmeurs. Python est un langage simple, facile à apprendre et permet une bonne réduction du coût de la maintenance des codes. Les bibliothèques (packages) python encouragent la modularité et la réutilisabilité des codes. Python et ses bibliothèques sont disponibles (en source ou en binaires) sans charges pour la majorité des plateformes et peuvent être redistribués gratuitement.

3 Présentation de l'implémentation

Nous utilisons un module d'intégration de texte Tensorflow-Hub pour créer notre classifieur de Fake News. Plus précisément, nous utilisons le module d'intégration de phrase nlm-en-dim128 (module d'intégration de texte préformé, présenté dans le chapitre 3), puis nous ajoutons notre propre couche de classification.

Notre implémentation est organisée en trois parties.

La première partie consiste à récupérer les News depuis la collection de test en utilisant la librairie Pandas. Les News récupérées contiennent en plus de l'énoncé de la News, d'autres métadonnées telles que l'auteur de la News, le sujet (ou thème) de la News ou encore l'affiliation de l'auteur. Du reste, la collection de test utilisée (présentée plus loin dans ce chapitre) classe

originellement les News en 6 classes. Nous constituons une nouvelle classification à la base de ces 6 classes comme suit :

- Classe originale : *True* → nouvelle classe : vraie News
- Classe originale : *mostly-true* → nouvelle classe : vraie News
- Classe originale : *Half-true* → nouvelle classe : vraie News
- Classe originale : *Barely-true* → nouvelle classe : fausse News
- Classe originale : *Pants-fire* → nouvelle classe : fausse News
- Classe originale : *False* → nouvelle classe : fausse News

Nous récupérons ainsi deux collections de données. Une collection qui est ensuite utilisée pour l'entraînement de notre modèle. Et une autre collection utilisée pour évaluer notre modèle.

La deuxième partie consiste à construire notre modèle et à l'entraîner avec les données d'entraînement ainsi récupérées en mettant en jeu les outils de prétraitement et de représentation de textes offerts par Tensorflow-Hub. Ces outils sont regroupés dans le module `nnlm-en-dim128`. Ce module permet comme son nom l'indique de représenter un texte avec un vecteur de 128 neurones en sortie. Notre modèle est construit en introduisant en plus deux couches de neurones cachés :

- Une première couche contenant 400 neurones cachés.
- Une deuxième couche contenant 100 neurones cachés.

Nous choisissons comme fonction d'activation la fonction RELU (présentée dans le premier chapitre). Pour l'entraînement, nous avons adopté d'utiliser une version optimisée de la Back-propagation (présentée dans le chapitre 1) et offerte par Tensorflow : il s'agit de l'*AdagradOptimizer*.

Un aperçu de notre code est donné ci-après.

```

main.py x
1 import tensorflow as tf
2 import tensorflow_hub as hub
3 import pandas as pd
4
5 # Je crée une fonction pour charger des données dans un tableau (DataFrame pandas)
6 def load_data(path):
7     # Voir le fichier README pour la structure du fichier
8     data = pd.read_csv(path,
9
10                        names=["id", "label", "statement", "subject",
11                              "speaker", "speaker-job", "state", "affiliation",
12                              "credit_0", "credit_1", "credit_2", "credit_3", "credit_4",
13                              "barely_true_count", "false_count", "half_true_count", "mostly_true_count", "pof_count",
14                              "context"],
15                        header=None,
16                        delimiter="\t")
17
18     for index, row in data.iterrows():
19         row['statement'] = row['statement'] + ' ' + str(row['subject']) + ' ' + str(row['affiliation']) + ' ' + str(row['speaker'])
20
21     # Je poste-traite les données
22     # Je suppose que j'ai 2 classes, que je map aux neurones en sortie (2 neurones, chaque neurone correspond à une classe)
23     data["label"] = data["label"].map({
24         "true": 0,
25         "mostly-true": 0,
26         "half-true": 0,
27         "barely-true": 1,
28         "pants-fire": 1,
29         "false": 1
30     })
31
32     return data
33
34 # CODE PRINCIPAL
35 load_data()

```

Figure 20 : Fragment de code

4 Évaluation

4.1 Collection de test

La collection de test que nous avons utilisée pour notre approche est une collection préexistante qui porte le nom Liar [32].

C'est une nouvelle collection de données de référence pour la détection de Fake News. Elle est parue dans le 53^{ème} congrès annuel de l'*Association for Computational Linguistics* (ACL 2017) qui est accessible au public pour les travaux de recherche. Cette collection compte 10269 news pour l'entraînement et 1283 news pour le test.

L'ensemble des News sont en anglais et contiennent des faits récents rapportés dans les médias et les réseaux sociaux. Exemple : « *Building a wall on the U.S.-Mexico border will take literally years.* » est classée comme vraie. Un autre exemple : « *Of \$60 billion Congress approved in response to Hurricane Sandy, only 10 percent was for disaster relief.* » est classée comme fausse. Les News sont donc, dans cette collection, des phrases assez courtes relatant un fait.

En plus de l'énoncé de la News, il y a d'autres métadonnées qui lui sont associées. Chaque News est alors représenté avec 14 colonnes :

- Colonne 1: Identifiant de contenu de la News (*ID of the statement*).

- Colonne 2: Classe ou étiquette de la News (*label*). Exemples : *true*, *mostly-true*, *half-true*, *barely-true*, *pants-fire* (très probablement fausse), *false*.
- Colonne 3: Le contenu (*the statement*).
- Colonne 4: Le(s) sujet(s) (*the subject(s)*). Exemples : *Ethics*, *jobs*, *Energy*, etc.
- Colonne 5: Auteur (*the speaker*). Exemple : *Barak-Obama*.
- Colonne 6: Titre du poste occupé par l'auteur (*the speaker's job title*).
- Colonne 7: Information sur l'endroit (état) où la News a été dite (si disponible) (*the state info*). Exemples : Illinois, Texas, etc.
- Colonne 8: Affiliation (*the party affiliation*). Exemples : *Republican*, *Democrat*, etc.
- Colonne 9: Nombre de News à peine vraies associées à l'auteur (*barely true counts*).
- Colonne 10: Nombre de News fausses associées à l'auteur (*false counts*).
- Colonne 11: Nombre de News à moitié vraies associées à l'auteur (*half true counts*).
- Colonne 12: Nombre de News presque vraies associées à l'auteur (*mostly true counts*).
- Colonne 13: Nombre de News très probablement fausses associées à l'auteur (*pants-on-fire count*).
- Colonne 14: le contexte de la News (*the context*). Exemple : « *a Democratic debate in Philadelphia* ».

4.2 Résultats

L'entraînement du modèle a été réalisé jusqu'à obtention d'une précision suffisante avec l'ensemble d'entraînement. Pour cela un nombre suffisant d'itération a été réalisé. Chaque itération correspond à une opération de Back-propagation sur l'ensemble des données d'entraînement. Ainsi, plus le nombre d'itération est grand plus la précision augmente. Dans ce qui suit, nous présentons les résultats avec prise en compte de différentes métadonnées.

4.2.1 Modèle 1 : Classification à base de contenu de la News et son thème

Le premier modèle est construit en prenant compte le contenu de la News et le thème de la News. Les résultats obtenus sont présentés dans ce qui suit.

Précision finale avec l'ensemble d'entraînement	Précision finale avec l'ensemble de test
0.99961	0.57143

Tableau 2 : Précisions finales avec l'ensemble d'entraînement et l'ensemble de test avec le modèle 1

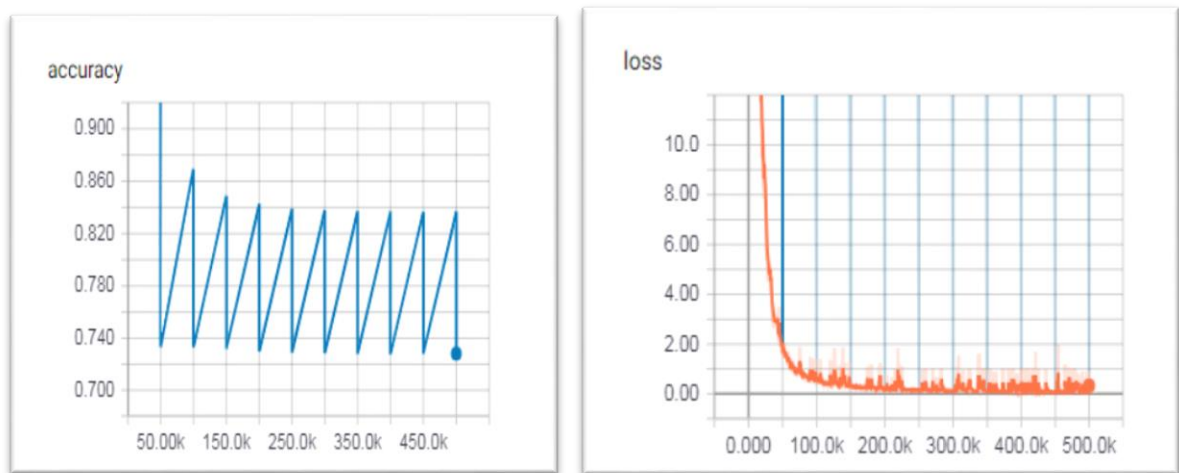


Figure 21 : Précision (accuracy) et erreur (loss) avec le modèle 1 en fonction du nombre d'itérations lors de l'entraînement

4.2.2 Modèle 2 : Classification à base de contenu de la News, son thème et l'auteur

Le deuxième modèle est construit en prenant compte le contenu de la News et le thème de la News et l'auteur de (ou personne ayant annoncé la News). Les résultats obtenus sont présentés dans ce qui suit.

Précision finale avec l'ensemble d'entraînement	Précision finale avec l'ensemble de test
0.99961	0.57774

Tableau 3 : Précisions finales avec l'ensemble d'entraînement et l'ensemble de test avec le modèle 2

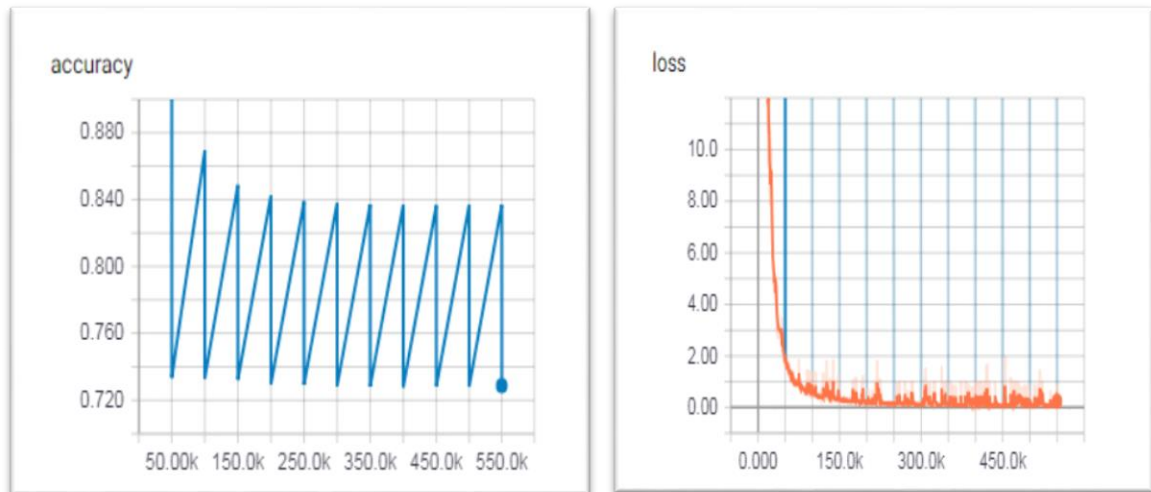


Figure 22 : Précision (accuracy) et erreur (loss) avec le modèle 2 en fonction du nombre d'itérations lors de l'entraînement

4.2.3 Classification à base de contenu du contenu de la News, son thème, l'auteur et son affiliation

Le deuxième modèle est construit en prenant compte le contenu de la News et le thème de la News et l'auteur de (ou personne ayant annoncé la News) ainsi que son affiliation. Les résultats obtenus sont présentés dans ce qui suit.

Précision finale avec l'ensemble d'entraînement	Précision finale avec l'ensemble de test
0.99961	0.56985

Tableau 4 : Précisions finales avec l'ensemble d'entraînement et l'ensemble de test avec le modèle 3

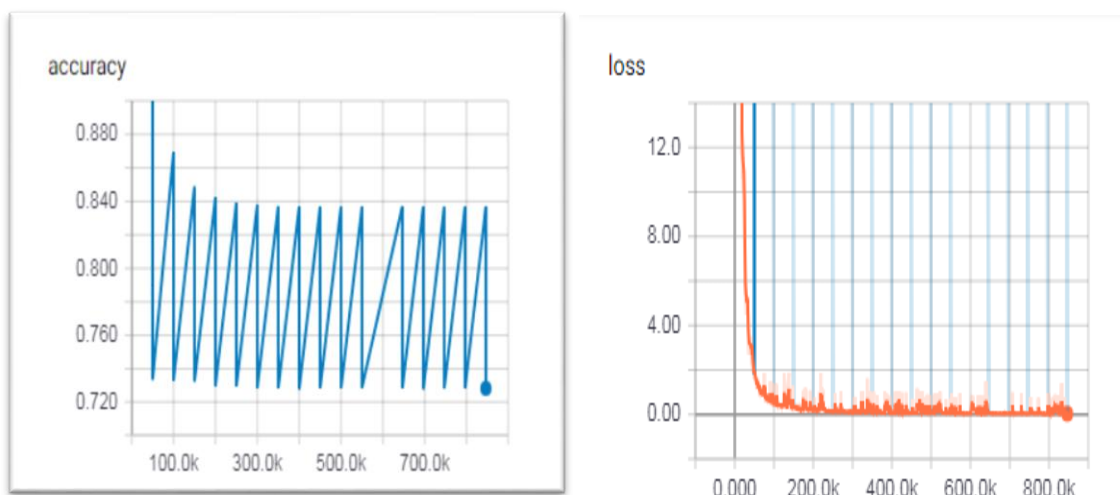


Figure 23 : Précision (accuracy) et erreur (loss) avec le modèle 3 en fonction du nombre d'itérations lors de l'entraînement

4.3 Discussion

Nous constatons que les résultats obtenus sont assez similaires pour les trois modèles. L'ajout de métadonnées influe peu la classification. De plus, les résultats sur la collection de test montrent un léger gain par rapport à une classification aléatoire (où 50% de bonne classification est espérée). En effet, près de 58% des News sont correctement classifiées avec le modèle 2.

Pour le reste nous remarquons que la précision de l'apprentissage et de test augmente avec le nombre d'itérations lors de l'entraînement. Ceci reflète qu'à chaque itération le modèle apprend plus d'informations par rapport à la collection.

	Nombre d'itérations (en milliers)	Précision finale avec l'ensemble d'entraînement	Précision finale avec l'ensemble de test	Temps d'exécution
Modèle 1	500	0.99961	0.57143	13min23
Modèle 2	550	0.99961	0.57774	13min43
Modèle 3	800	0.99961	0.56985	14min36

Tableau 5 : Tableau comparatifs de résultats

Le modèle 2 a présenté les meilleurs résultats. Ainsi donc, la prise en compte de l'affiliation de l'auteur semble réduire la qualité des résultats. La prise en compte de l'auteur seul améliore de façon notable la précision avec l'ensemble de test.

5 Conclusion

Dans ce chapitre, nous avons présenté les logiciels et bibliothèques utilisés pour notre implémentation. Ensuite, nous sommes passés à la description de l'implémentation réalisée. Nous avons en particulier présenté les bibliothèques et outils utilisés ainsi que notre code.

L'évaluation réalisée montre un certain intérêt de notre approche. Cependant, le classifieur doit être d'autant plus amélioré pour envisager de l'utiliser dans une situation réelle. En effet, la précision réalisée, bien que plus élevée qu'une classification aléatoire reste faible (58%).

Conclusion générale

1 Synthèse

Le domaine de la classification a particulièrement progressé ces dernières années grâce aux techniques de l'apprentissage automatique. Mais les techniques de classification des Fake News à base d'apprentissage automatique sont loin d'être performantes à 100%. Donc, il est très important de continuer de progresser dans le domaine de traitement linguistique du texte, afin d'arriver à une représentation textuelle manipulable par les algorithmes de classification, mais aussi de construire et de modéliser des classifieurs performants.

A travers ce mémoire, nous avons présenté d'une façon complète les différentes notions nécessaires pour la compréhension de notre sujet. Nous avons organisé le mémoire en quatre chapitres, chacun traitant une partie de l'objectif global. Le dernier chapitre résume les différents résultats expérimentaux.

Les différentes expérimentations réalisées ont montré l'intérêt des classifieurs adoptés. Aussi, nous avons montré l'intérêt de la prise en compte des métadonnées liées aux News (auteur, affiliation, etc.) lors de la classification.

2 Perspectives

Nous envisageons plusieurs suites à nos travaux

- La première perspective serait de tester d'autres collections de test, notamment celles portant sur des langues différentes (autre que l'anglais).
- Une deuxième perspective serait d'utiliser un module plus représentatif (nnlm-512).
- Une autre perspective consiste à réaliser une interface graphique pour notre classifieur.

Bibliographie

-
- [1].....McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133.
- [2].....Apprentissage Automatique. En ligne:
https://fr.wikipedia.org/wiki/Apprentissage_automatique
- [3].....Définition: Apprentissage automatique. En ligne:
<http://www.psychomedia.qc.ca/lexique/definition/apprentissage-automatique>
- [4].....Cornuéjols, A., & Miclet, L. (2011). *Apprentissage artificiel: concepts et algorithmes*. Editions Eyrolles.
- [5].....Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4), 279-292.
- [6].....Apprentissage par transfert:une nouvelle approche pour les réseaux de neurones profonds, 2018
- [7].....S.Aissaoui, « Apprentissage automatique et sécurité des systèmes d'informations ». Thèse de Doctorat (2008). Oran.
- [8].....Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.
- [9].....Laroum, S., Béchet, N., Hamza, H., & Roche, M. (2010). Classification automatique de documents bruités à faible contenu textuel. *Revue des Nouvelles Technologies de l'Information*, (spécial: Fouille de Données Complexes), 25.
- [10].....Parizeau, M. (2004). Réseaux de neurones. GIF-21140 et GIF-64326, 124.
- [11].....Szücs, R., Verleysen, K., Duchateau, G. S., Sandra, P., & Vandeginste, B. G. (1996). Analysis of phospholipids in lecithins comparison between micellar electrokinetic chromatography and high-performance liquid chromatography. *Journal of Chromatography A*, 738(1), 25-29.
- [12].....Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.
- [13].....Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1), 1-47.
- [14].....Mattallah Hocine, « Classification Automatique De Textes Approche Orientee Agent ». Thèse de Magistère (2011). Tlemcen.

-
- [16]Introduction à l'apprentissage automatique. En ligne :
<http://vision.gel.ulaval.ca/~cgagne/enseignement/apprentissage/A2016/presentations/ar-sem01-introduction.pdf>
- [17]Furnkranz, J., Mitchell, T., & Riloff, E. (1998). A case study in using linguistic phrases for text categorization on the WWW. In Working Notes of the AAAI/ICML, Workshop on Learning for Text Categorization (pp. 5-12).
- [18]Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130-137.
- [19]Ignat, C., & Rousselot, F. (2006). Représentation de Textes à l'Aide d'Étiquettes Sémantiques dans le Cadre de la Classification Automatique. *Revue roumaine de linguistique*, 51(3-4), 421-439.
- [20]Jalam, R., & Chauchat, J. H. (2002). Pourquoi les n-grammes permettent de classer des textes? Recherche de mots-clefs pertinents à l'aide des n-grammes caractéristiques. In 6th International Conference on Textual Data Statistical Analysis, France (pp. 381-390).
- [21]Catégorisation automatique de textes et Cooccurrence de mots provenant de documents non étiquetés. Mémoire. Québec, Université Laval, Canada.2005
- [22]Wiener, E., Pedersen, J. O., & Weigend, A. S. (1995, April). A neural network approach to topic spotting. In Proceedings of SDAIR-95, 4th annual symposium on document analysis and information retrieval (Vol. 317, p. 332).
- [24]Les « fake news » : c'est quoi ? En ligne : <https://www.justaskgemalto.com/fr/les-fake-news-cest-quoi/>
- [26]L'objectif caché derrière les sites de Fake News. En ligne :
<http://quebec.huffingtopost.ca/découvrir>
- [27]Fake News : une nouvelle pratique d'influence difficilement maîtrisable. Rida Ahib, Rachad Jebari, Nicolas Lefort, Caroline Phillips et Carlo Valle, 2018.
- [28]Dreyfus G. «Les réseaux de neurones » *Mécanique Industrielle et Matériaux* », n051, 1998.
- [29]Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb), 1137-1155.
- [30]Tensorflow (site officiel). En ligne: <https://www.tensorflow.org/>
- [31]La bibliothèque logicielle Pandas. En ligne: <https://pandas.pydata.org>

-
- [32].....Wang, W. Y. (2017). " Liar, liar pants on fire": A new benchmark dataset for fake news detection. ArXiv preprint arXiv:1705.00648
- [33].....NumPy. En ligne: <https://fr.wikipedia.org/wiki/NumPy>
- [34].....TensorBoard: Visualizing Learning and TensorFlow. En ligne: https://www.tensorflow.org/guide/summaries_and_tensorboard
- [35].....Page github de tensorflow-hub. En ligne: <https://github.com/tensorflow/hub>
- [36].....Le langage Python. En ligne: [https://fr.wikipedia.org/wiki/Python_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage))
- [37].....Partie I : que sont les métadonnées ? En ligne : <https://www.icann.org/news/blog/partie-i-que-sont-les-metadonnees>
- [38].....Blayo, F., & Verleysen, M. (1996). Les réseaux de neurones artificiels.
- [39].....MEDJILI, F. (2010). Modélisation par Réseaux de Neurones Artificiels (RNA) et commande Prédictive non linéaire d'une station de production d'eau froide (Doctoral dissertation, Université Badji Mokhtar de Annaba).
- [40].....Rivals, I., Personnaz, L., Dreyfus, G., & Ploix, J. L. (1995). Modélisation, classification et commande par réseaux de neurones: principes fondamentaux, méthodologie de conception et illustrations industrielles. Les réseaux de neurones pour la modélisation et la commande de procédés, JP Corriou, ed. (Lavoisier Tec & Doc, 1995).