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

UNIVERSITE SAIDA - Dr. MOULAY Tahar

Faculté : Technologie

Département : Informatique

MEMOIRE DE MASTER

Option : Sécurité Informatique et Cryptographie (SIC)

Thème

Détection de spam avec l'apprentissage profond

Présenté par :

SARIA Redouane
HACHEROUF Nour El Islam

Encadré par :

BOUDIA M.Amine

Promotion : Septembre 2020

Remerciement

Nous remercions tout d'abord Dieu, le miséricordieux de nous avoir aidé et donné la force et le courage.

Nous tenons à remercier notre encadreur Monsieur ***BOUDIA Mohamed amine*** *qui s'est toujours montré à l'écoute et pour sa disponibilité tout au long de la réalisation de ce mémoire.*

Nous tenons aussi à exprimer nos remerciements aux membres du jury qui ont accepté d'évaluer notre travail.

à remercier sincèrement Mademoiselle ***MEKKI Nour*** *pour le soutien et l'aide qu'elle n'a jamais manqué de nous apporter durant l'élaboration de ce travail.*

Enfin, nous adressons nos plus sincères remerciements à tous ceux qui de près ou de loin ont apporté un effort pour l'élaboration et la mise en forme de ce modeste travail.

Merci à tous.

DEDICACES

Je dédie ce modeste travail à : A mes parents .Aucun hommage ne pourrait être à la hauteur de l'amour Dont ils ne cessent de me combler. Que dieu leur procure bonne santé et longue vie.

A celui que j'aime beaucoup et qui m'a soutenue tout au long de ce projet : mon frére ***TAAMMA Rafik Baghdad****, et bien sur A mes frères* ***Houcine*** *et* ***Hacen, Mohamed, Toufik, Baghdad, abdelatif, oussama*** *sans oublié ma grand-mère et mes beaux-parents que j'aime.*

A toute ma famille, et mes amis, A mon binôme ***Nour EL Islam*** *et toute la famille* ***SARIA****. Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.*

REDOUANE

DEDICACES

JE DEDIE CE MEMOIRE A …

Mon père

Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous.

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être.

Ma très chère mère

Je te dédie ce travail en témoignage de mon profond amour. Puisse dieu, le tout puissant, te préserver et t'accorder sante, longue vie et bonheur.

les fleurs de notre maison : Yassemin, iness

Je vous dédie ce travail avec tous mes vœux de bonheur, de sante et de réussite.

A tous les membres de ma famille hacherouf, petits et grands

Veuillez trouver dans ce modeste travail l'expression de mon affection

A mes chère professeures

Un remerciement particulier et sincère pour tous vos efforts fournis. Vous avez toujours été présents.

Que ce travail soit un témoignage de ma gratitude et mon profond respect.

Merci…islam

Table des matières

Table des matières

Table des matières

Table des matières

Liste des figures

Liste des tableaux

RESUME

Le courrier électronique rend vraiment service aux usagers, c'est un moyen rapide et économique pour échanger des informations. Cependant, les utilisateurs se retrouvent assez vite submergés de quantités de messages indésirables appelé aussi spam. Le spam est rapidement devenu un problème majeur sur Internet. Dans cet article, nous proposons une nouvelle collecte de spam SMS réel, public et non encodé qui est la plus importante à notre connaissance. De plus, nous comparons les performances obtenues par plusieurs méthodes d'apprentissage en profondeur établies

Dans le cadre de notre travail, la classification des courriers électronique est effectuée à l'aide de deux architectures d'apprentissage profond : Les réseaux de neurones convolutifs (CNN), Les réseaux de neurones récurrents(RNN). L'efficacité de ces classificateurs est testée avec des différentes représentations on utilisant le corpus smsSpamCollection. Les résultats des tests montrent que CNN est plus performant par rapport aux RNN.

Mots-clés : Spam, architecture d'apprentissage profond, réseaux de neurones convolutifs (CNN), réseaux de neurones récurrents (RNN).

ABSTRACT

The e-mail really makes service to users; it is a fast and economical way to exchange information. However, users find themselves quickly overwhelmed with amounts of unwanted messages called spam. Spam has quickly become a major problem on the Internet. In this paper, we offer a new real, public and non-encoded SMS spam collection that is the largest one as far as we know. Moreover, we compare the performance achieved by several established deep learning methods.

As part of our work, the classification of electronic mails is carried out using two deep learning architectures: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN). The efficiency of these classifiers is tested with different representations using the smsSpamCollection corpus. Test results show that CNN performs better compared to RNN.

Keywords: Spam, deep learning architectures, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN).

INTRODUCTION GENERALE

Le courrier électronique (ou courriel, email) est un des services les plus utilisés sur internet, il est sans doute la technique qui a changé nos habitudes à une grande échelle. La croissance de l'Internet est reliés directement à l'importance du courriel, car plusieurs sites web lui sont maintenant consacrés, et presque tous les gens qui ont accès à internet ont au moins une adresse de courrier électronique qu'ils vérifient quotidiennement, ce qui explique les milliards des courriels qui s'envoient et sont reçus chaque jour.

Aujourd'hui, le courriel rend vraiment service aux usagers, c'est un moyen rapide et économique pour échanger des informations. Si nous comparons le courrier électronique aux autres moyens de communication, (par écrit, téléphone), nous nous apercevons que les avantages des courriels surpassent ses inconvénients. Sa force réside dans le médium du transport des messages, la rapidité avec laquelle circulent les courriels, l'économie, la disponibilité en tout temps indépendamment du décalage horaire et à la possibilité de les envoyer à plusieurs personnes en même temps. La nature informatique de ces courriels offre des avantages incomparables, dont l'envoi des documents électroniques par attachement, l'archivage des messages est beaucoup plus facile à effectuer qu'avec les communications écrites ou par téléphone, ainsi que, le courrier électronique permet d'effectuer un traitement rapide, efficace et automatique sur les messages comme la recherche par mots clés, le tri automatique par sujet.

Cependant, les utilisateurs se retrouvent assez vite submergés de quantités de courriers électroniques indésirables ou non sollicités appelés aussi spam. En effet, le spam est rapidement devenu un problème majeur sur Internet.

Le spam est un phénomène mondial et massif. Selon la CNIL (La Commission Nationale de l'Informatique et des Libertés), le spam est défini de la manière suivante : « Le "spamming" ou "spam" est l'envoi massif de courriers électroniques non sollicités, à des personnes avec lesquelles l'expéditeur n'a jamais eu de contact et dont il a capté l'adresse électronique de façon irrégulière. », Il existe de nombreuses techniques contre le spam qui peuvent être divisées en deux groupes. Le premier contient les solutions basées sur l'entête du message électronique telles que les listes noires et les listes blanches. Le deuxième groupe de solutions contient celles qui sont basées sur le contenu textuel du message telles que le filtrage basé sur l'apprentissage automatique.

Il existe de nombreux travaux qui traitent le problème de filtrage de spam en utilisant des méthodes d'apprentissage automatique. Le filtrage de spam basé sur le contenu textuel des messages peut être considéré comme un exemple de classification de textes qui consiste en l'attribution de documents textuels à un ensemble de classes prédéfinies. Le but d'un système de classification est d'effectuer la tâche de classification et de le faire avec un degré raisonnable d'exactitude. Il existe aujourd'hui une liste plutôt longue de classifieurs développés autour des algorithmes.

INTRODUCTION GENERALE

Dans notre travail, nous étudions la détection des spams en utilisant deux modèles d'apprentissage profond : le RNN (*récurrent neural network*) qui est largement utilisé dans le domaine de traitement de texte et le CNN (convolutional neural networks). Les deux a donné de très bons résultats. Une étude comparative est faite entre les deux architectures proposées et d'autres travaux traitant le même domaine.

Le travail est réparti en 3 chapitres :

Le premier chapitre : dans ce chapitre nous abordons le sujet de détection des spams.

Le deuxième chapitre : présent l'apprentissage profond et les réseaux de neurones convolutifs.

Le troisième Chapitre : le chapitre présente les différents outils qui vont servir à l'implémentation du projet, ainsi que les différentes architectures proposées et les discussions sur les résultats.

Chapitre 1 - Détection Des Spams

1.1. Introduction

Le spam est un grand problème pour les internautes. Les augmentations récentes du taux de spam ont causé une grande inquiétude parmi la communauté Internet. De nombreuses solutions avaient été suggérées pour résoudre le problème.

Dans ce chapitre, nous présentons tout d'abord les débuts du spam, ses objectifs, ses contenus, ses impacts et les différentes techniques utilisées pour détecter ce type de courriels.

1.2. Naissance et débuts du spam

1.2.1. Origine du mot spam

En 1937 La société Hormel Foods organise un concours pour trouver un nouveau nom pour leur jambon épicé, Ce nom doit être aussi caractéristique que le goût du produit « Spiced Ham » et qui propose « Spam » pour ce produit, fut donc la marque retenue.

Cette viande précuite en boîte souvent synonyme de mauvaise nourriture a été largement utilisée par l'intendance des forces armées américaines pour la nourriture des soldats pendant la Seconde Guerre mondiale et sera introduite dans diverses régions du monde à cette occasion. [1]

1.2.2. Définition du spam

Le spam est un message électronique non sollicité, envoyé massivement à un grand nombre de destinataires, à des fins publicitaires ou malveillantes. [1]

Le terme spam est aussi utilisé pour désigner le même type de message transmis par d'autres moyens de communication électroniques tels que les messageries instantanées, les blogs, les forums, et plus récemment, des réseaux de téléphonie mobile, via les SMS ou MMS. Même si le moyen de communication est différent, les techniques d'envoi et de détection restent relativement similaires.

Le premier spam (Figure 1.1) date du 3 mai 1978. Ce jour là, sur le réseau ARPANET , Gary Thuerk, commercial de la société informatique DEC3, invitait par e-mail 393 personnes à découvrir sa nouvelle machine, le 2020.

Hormel Foods : fabricant de viande en conserve

ARPANET : est le premier réseau à transfert de paquets développé aux États-Unis

DEC : Digital Equipment Corporation

Le message se présentait ainsi :

Mail-from: DEC-MARLBORO rcvd at 3-May-78 0955-PDT
Date: 1 May 1978 1233-EDT
From: THUERK at DEC-MARLBORO
Subject: ADRIAN@SRI-KL

WE INVITE YOU TO COME SEE THE 2020 AND HEAR ABOUT THE DECSYSTEM-20 FAMILY AT THE TWO PRODUCT PRESENTATIONS WE WILL BE GIVING IN CALIFORNIA THIS MONTH. THE LOCATIONS WILL BE:

TUESDAY, MAY 9, 1978 – 2 PM
HYATTHOUSE (NEAR THE L.A. AIRPORT)
LOS ANGELES, CA

THURSDAY, MAY 11, 1978 – 2 PM
DUNFEY'S ROYAL
COACH SAN MATEO, CA
(4 MILES SOUTH OF S.F. AIRPORT AT BAYSHORE, RT 101 AND RT 92)

A 2020 WILL BE THERE FOR YOU TO VIEW. ALSO TERMINALS ON-LINE TO OTHER DECSYSTEM-20 SYSTEMS THROUGH THE ARPANET. IF YOU ARE UNABLE TO ATTEND, PLEASE FEEL FREE TO CONTACT THE NEAREST DEC OFFICE FOR MORE INFORMATION ABOUT THE EXCITING DECSYSTEM-20 FAMILY.

Figure 1.1 Le premier spam

Ce message indésirable n'était hélas que le premier d'une longue série. Le spam était né. [2]

1.3. Objectifs et statistiques sur les spam

Au départ, le spam visait principalement des objectifs publicitaires. Aujourd'hui, il s'est considérablement développé, diversifié et complexifié, pour atteindre de plus en plus souvent des objectifs malveillants. En effet, Le spam s'est non seulement développé en termes de volume, mais également en termes de contenu (voir figure 1.2). Aujourd'hui, les objectifs des spam sont très variés en voici une liste non exhaustive :

1.3.1. Hameçonnage (ou phishing) :

L'objectif est de réussir à se faire passer pour un organisme connu par l'utilisateur, dans le but de lui voler des informations à caractère confidentiel. Par exemple, on reçoit un mail provenant "apparemment" de notre banque, ou d'un autre site où l'on dispose d'informations personnelles. dans ce mail, il est demandé de cliquer sur un lien (pour des motifs divers, réactualisation, etc.), après avoir cliqué sur ce lien, une page web s'affiche... sur laquelle il est demandé de rentrer ses coordonnées bancaires ou toute autre information personnelle. Parmi les sites Top les plus contrefaits pour les attaques de phishing, on retrouve eBay, Paypal et Bank of America. [3]

1.3.2. Publicité :

L'objectif est de vanter les mérites d'un produit quelconque. Il s'agit par exemple de produits pharmaceutiques, de produits de luxe, de logiciels divers et variés, de jeux d'argent. Ils peuvent également soutenir-agate idées politiques, culturelles ou religieuses et organisations.

1.3.3. Scam :

Il s'agit d'une attaque basé sur la naïveté des destinataires dans le but de leur soutirer de l'argent. L'exemple le plus courant est le scam nigérien: un dignitaire d'un pays d'afrique vous demande de servir d'intermédiaire pour une transaction financière importante, en vous promettant un bon pourcentage de la somme. Pour amorcer la transaction, il vous faut donner de l'argent. [4]

1.3.4. Canular :

L'objectif est de faire circuler une information semblant très sensible, souvent avec un caractère d'urgence : fausse alerte de virus, fausse alerte de contamination potentielle, chaine de solidarité….. Par exemple : « un nouveau virus très dangereux se propage, il faut faire circuler l'information » ; « des sousvêtements sont infectés par une dangereuse bactérie ».

1.3.5. Malware :

Est un logiciel conçu pour infiltrer ou endommager un système informatique. Il est communément pris pour contenir des virus informatiques, vers, chevaux de Troie, spywares et adwares. Ce type de logiciel est souvent envoyé en tant que non suspect d'une pièce jointe. Lorsque l'utilisateur ouvre le fichier, le logiciel malveillant s'installe. L'interdépendance entre les spams et les logiciels malveillants a évolué Spam logiciels malveillants propagation des e-mails, les logiciels malveillants est utilisé pour infecter un hôte de sorte que l'hôte peut être contrôlé à distance et utilisé pour l'envoi de plus de spams. Ces hôtes infectés sont désignées comme des «ordinateurs zombies». Beaucoup de gens croient que la plupart des spams sont envoyés par des botnets, qui constituent un réseau de PC zombies. [5]

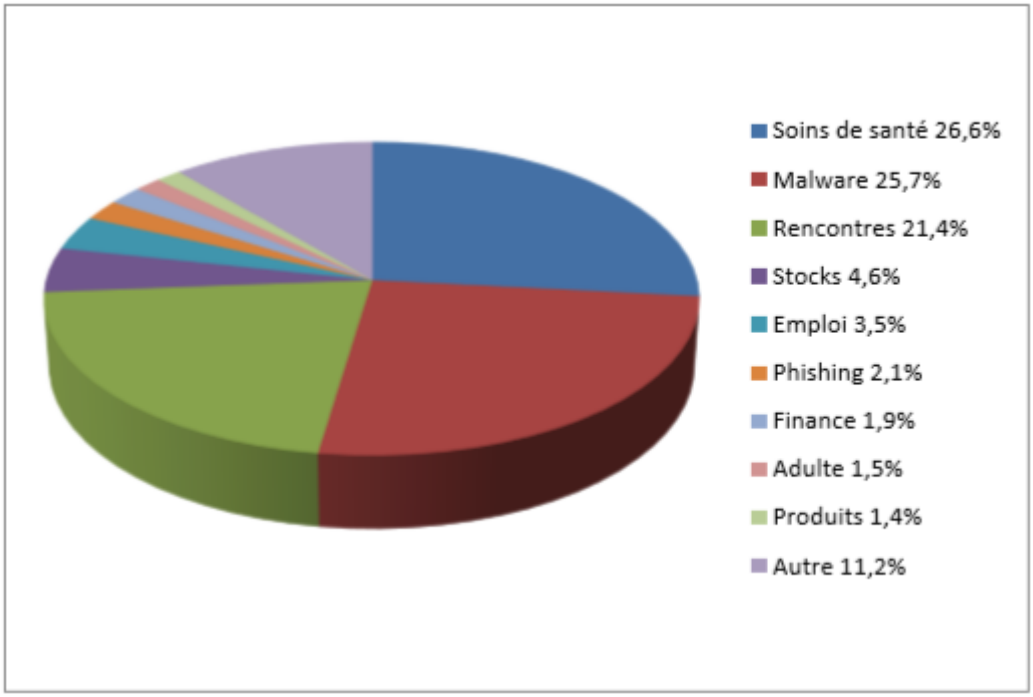


Figure 1.2 Répartition des spam par contenu [6]

On a quelques statistiques sur le taux global de spam entre les années 2012 et 2017 présenté dans la figure 1.4. Dans la dernière période il a été constaté que le spam représentait 55% de tous les messages électroniques, comme au cours de l'année précédente.

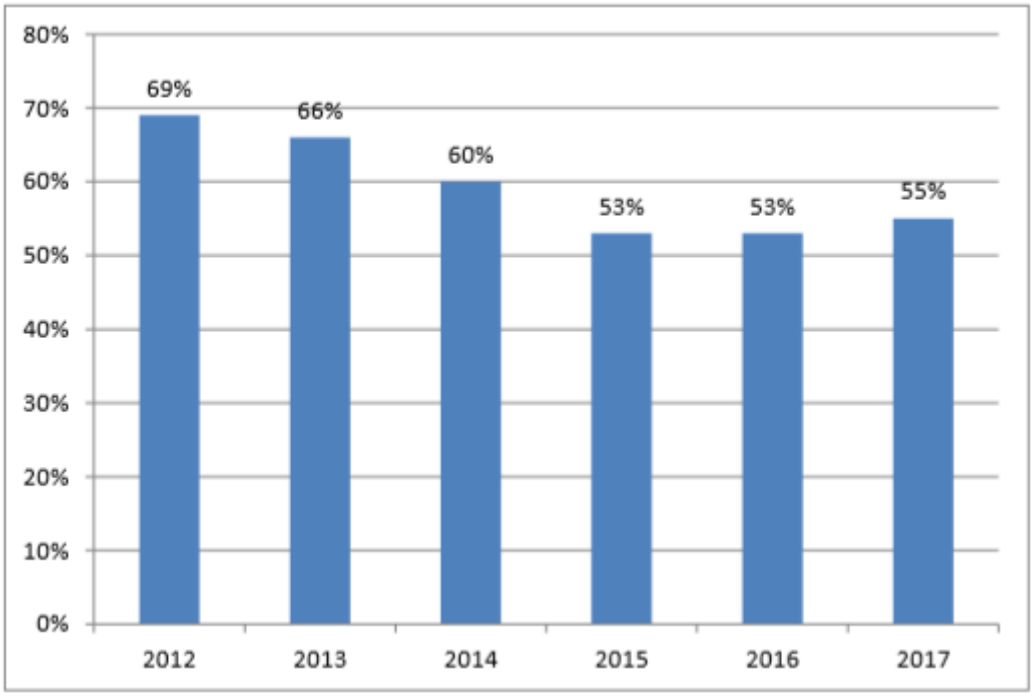


Figure 1.3 développement de spam en termes de volume [6]

1.4. Impactes du spam sur les utilisateurs et les fournisseurs

Dans cette section, nous présentons les effets du spam, au niveau des utilisateurs, entreprises et FAI. [7]

1.4.1. Perte de temps :

- Encombrement anormal des boîtes aux lettres.
- Suppression des courriels indésirables.
- Configuration et maintenance des filtres.
- Consultation des courriels rejetés pour y détecter les bons à cause du risque de passer à côté d'emails importants mal catalogués par les outils de détection anti-spam

1.4.2. Perte de bande passante et d'espace disque :

- Spécialement pour les utilisateurs de modems.
- Les pièces jointes des virus et spam peuvent être grands.

1.4.3. Pertes financières non négligeables aux niveaux des entreprises et FAI :

- une augmentation des coûts de gestion opérationnelle et support lié à la gestion anti spam.
- perte de productivité des salariés,

Selon une étude, le spam aurait coûté environ 712 $ par employé et par an aux entreprises. À ce chiffre, il faut rajouter 113 à 183 $ par employé et par an pour la gestion des emails en quarantaine.

1.5. Techniques de détection ou filtrage du spam :

Plusieurs techniques de lutte contre le spam sont possibles et peuvent être cumulées : analyse statistique (filtre bayésien), filtrage par mots clés, listes blanches, listes noires. Ces techniques de lutte doivent s'adapter en permanence car de nouveaux types de spam réussissent à les contourner.

Deux solutions de détection de spam sont envisageables : la détection au niveau du serveur et la détection au niveau de l'utilisateur final.

Ces outils peuvent être divisés en deux groupes : le filtrage d'enveloppe, et le filtrage de contenu.

1.5.1. Filtrage d'enveloppe :

Ce type de filtrage s'applique uniquement sur l'en-tête4 du message, qui contient souvent assez d'informations pour pouvoir distinguer un spam. Cette technique appliquée au niveau du serveur FAI présente l'avantage de pouvoir bloquer les courriels avant même que leur corps ne soit envoyé, ce qui diminue grandement le trafic sur la passerelle SMTP.

Dans cette catégorie, nous trouvons les techniques suivantes :

1.5.1.1. Filtrage par listes noires :

Ces listes consistent à pré-déclarer une liste de « mauvais expéditeurs », (adresses emails, noms des domaines, pays, adresse IP) ou un message envoyé, desquelles le destinataire refuse de recevoir des emails ou des messages. Ces listes peuvent être :

- créées par l'administrateur ou l'utilisateur.

- téléchargées via le web (cela nécessite une mise à jour très régulière pour un filtrage optimisé)
- consultées en temps réel sur le web (RBL, Real Time Blackhole List).

Pour contourner ces listes les spammeurs, changent très fréquemment leurs adresses d'expédition (email, ou message). [26]

1.5.1.2. Filtrage par listes blanches :

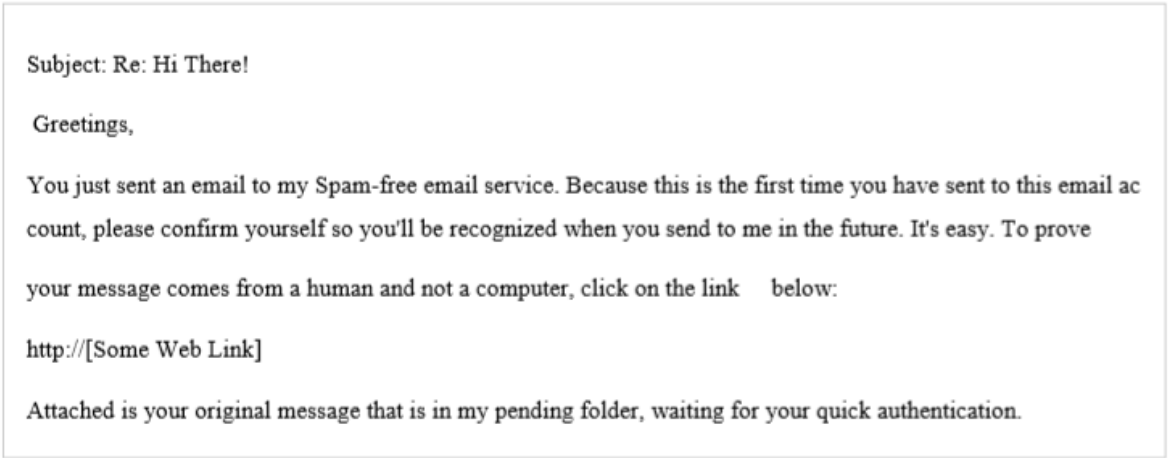
Ces listes consistent à pré-déclarer une liste de (adresses emails, noms des domaines, adresse IP) sûres desquels le destinataire accepte de recevoir des emails. Par défaut très peu d'hôtes sont considérés comme sûrs car leurs adresses pourraient être usurpées par les spammeurs. Tout comme la liste noire, la liste blanche a également besoin d'une mise à niveau continue et de rafraîchissement. [27]

1.5.1.3. Filtrage par liste grise :

La liste grise est un mixte entre la liste blanche et la liste noire. Ce qui se produit est qu'à chaque fois qu'une boîte aux lettres donnée reçoit un email d'un contact inconnu, cet email est suspendu avec un message de réponse automatique contenant un lien permettant de valider l'envoi. Ceci à pour but de détecter les robots, les spammeurs ne se rendront pas compte qu'ils doivent émettre une validation afin que le message soit accepté.

Dans le cas d'un réel email attendu et que l'expéditeur n'est pas énumérée dans l'une ou l'autre des listes noire et blanche, alors il sera positionné en liste grise. Si l'expéditeur satisfait la demande de confirmation (souvent un lien Web à cliquer), il obtiendra alors le passage à liste blanche et ses messages vous seront acheminés. C'est en fait l'ouverture dynamique de la liste blanche.

Par exemple : la figure suivante présente une réponse de cette technique.


Subject: Re: Hi There!

Greetings,

You just sent an email to my Spam-free email service. Because this is the first time you have sent to this email ac count, please confirm yourself so you'll be recognized when you send to me in the future. It's easy. To prove

your message comes from a human and not a computer, click on the link below:

http://[Some Web Link]

Attached is your original message that is in my pending folder, waiting for your quick authentication.

Figure 1.4 Exemple sur une réponse de technique de la liste grise [26]

1.5.1.4. Filtrage par vérification du domaine :

Les destinataires sont configurés de sorte qu'ils n'acceptent que les messages provenant de domaines spécifiques. Les e-mails dont les domaines ne sont pas mentionnés ne seront pas reçus. De cette façon, beaucoup de spam est bloqué. [28]

1.5.2. Filtrage du contenu

Ce type de filtrage se fait au niveau de l'utilisateur où son contenu est analysé pour détecter les spam qui ont réussi à passer à travers le filtre d'enveloppe.

Dans cette catégorie nous trouvons les techniques suivantes :

1.5.2.1. Filtrage par mots clés :

L'administrateur doit indiquer la liste des mots clés à détecter afin de déterminer qu'un mail est un Spam. Par exemple, tous les emails qui contiennent les mots : viagra, argent, money, drogue seront détectés comme Spam.

Ce filtre se base sur les mots clé inclus dans les mails. L'analyse est très rapide, mais peux efficace. Car cela demande un suivi manuel et les Spammeurs font varier les mots clé afin d'éviter ce filtre. Par exemple, on retrouve M.O.N.E.Y. ou encore m*o*n*e*y. [29]

1.5.2.2. Filtrage par caractères :

Il s'agit de bloquer les emails qui contiennent certains caractères ou police de caractère, ou certaines langues utilisées dans ces emails.

1.5.2.3. Filtrage d'image :

Il s'agit d'analyser les images obtenues dans les messages au niveau des propriétés du fichier image (format, taille du fichier, taille d'image) que du contenu de l'image (couleurs, test de pixels,…).

1.5.2.4. Filtrage d'URL :

Ceci consiste à vérifier les liens hypertextes inclus dans les messages auprès d'une base de données de « mauvais URL » préenregistrés, ou via la consultation en temps réel des listes noires disponibles sur le web. Des tentatives de masquage du lien hypertexte sont des fois utilisées par des spammeurs pour empêcher l'analyse par le filtrage d'URL.

1.5.2.5. Filtres bayésiens :

L'approche d'apprentissage automatique le plus connu dans le filtrage des spams est le classificateurs Bayes naïfs, classificateur Naïve Bayes est un classificateur probabiliste. En bref, il calcule et utilise la probabilité de certains mots / expressions apparaissant dans les exemples les plus connus (messages) afin de classer de nouveaux exemples (messages). Naïve Bayes a été montré pour être très bien réussi à catégoriser les documents texte. Filtres bayésiens (méthode statistique) Filtres travaillé en analysant les mots du message à l'intérieur d'un e-mail pour calculer la probabilité que le message est un spam ou non. Le calcul basé sur des mots qui déterminent que le message est un spam et les mots qui déterminent que le message n'est pas du spam.

1.5.2.6. Machine à Vecteurs de Support (SVM) :

Machine à Vecteurs de Support (SVM) ont eu du succès dans le classement des documents texte. SVM a donné lieu à une recherche importante dans les appliquer à filtrage de spam. SVM sont des méthodes à noyaux dont l'idée centrale est d'intégrer les données représentant les documents texte dans un espace vectoriel. SVM tenter de construire une séparation linéaire entre deux classes dans cet espace vectoriel.

Une machine à vecteurs de support est un classifieur linéaire binaire à marge maximale. Il peut être interprété comme trouver un hyperplan dans un espace de

caractéristiques linéairement séparables qui sépare les deux classes avec une marge maximum. Les instances les plus proches de l'hyperplan sont connues comme les « vecteurs de support » car ils soutiennent l'hyperplan des deux côtés de la marge.

SVM a été rapporté significative des performances sur le problème de la catégorisation de textes avec de nombreuses fonctionnalités pertinentes. SVM a également été appliquée au filtrage anti-spam. [5]

1.6. Conclusion :

Dans ce chapitre en à présente de définitions de spam, ses objectifs et impacts ainsi les différents approches a battant de spam en peu divise en deux catégories : Le premier contient les solutions basées sur l'en-tête du message électronique telles que les listes noires, blanches et grises. Le deuxième groupe de solutions contient celles qui sont basées sur le contenu textuel du message telles que le filtrage basé sur l'apprentissage profond (deep learning)

Chapitre 2 – Apprentissage Profond (Deep Learning)

2.1 Introduction :

Né dans les années 1950 avec les travaux d'Alan Turing, puis de John McCarthy et Marvin Lee Minsky, l'intelligence artificielle est l'un des sujets de bouleversements majeurs qui affectent notre époque. L'intelligence artificielle, souvent abrégée avec le sigle IA, est définie par l'un de ses créateurs, Marvin Minsky (Minsky, 2007), comme : " la construction de programmes informatiques qui s'adonnent a` des tâches qui sont pour l'instant, accomplies de façon plus satisfaisante par des ˆêtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critiquée. Il existe deux types d'intelligence : l'intelligence artificielle faible et forte.

1. L'IA forte : elle fait référence à une machine qui soit capable d'éprouver une réelle conscience de soi, ressentir de vrais sentiments et comprendre ce qui la pousse a` faire telle ou telle action : on recherche à reproduire à l'identique le fonctionnement du système cognitif humain. On parlera alors de cognition artificielle (La machine pense !).
2. L'IA faible : elle est en quelque sorte plus raisonnable que celle de l'IA forte. Elle considère qu'un programme peut-être capable de raisonner, d'apprendre et même de résoudre des problèmes, mais cette fois, le programme simule l'intelligence, semble agir comme s'il était intelligent.

DL est l'une des raisons qui ont conduit les récents mouvements et progrès de l'IA, et c'est la principale cause qui fait penser que finalement, il existe une possibilité pour l'IA de devenir plus réaliste. Alors, qu'est-ce que DL?

Selon les fondateurs Yann LeCun, Yoshua Bengio Geoffrey Hinton dans [8] :
"L'apprentissage profond permet aux modèles informatiques composés de plusieurs couches de traitement d'apprendre des représentations de données avec plusieurs niveaux d'abstraction."

Autre définition par les auteurs dans [9] :
"L'apprentissage profond est une classe de techniques d'apprentissage machine, où l'information est traitée en couches hiérarchiques pour comprendre les représentations et les caractéristiques des données dans des niveaux de complexité croissante."

En d'autres termes, DL est un sous-ensemble des méthodologies et techniques de ML qui utilisent le réseau neuronal artificiel (ANN). C'est l'adaptation des réseaux neuronaux qui imite la structure du cerveau humain. La force de DL réside dans le fait que la machine peut extraire des caractéristiques et apprendre toute seule, indépendamment de l'intervention d'un expert. Il a été appliqué dans de nombreux domaines différents (traitement des images, textes, paroles et vidéos). Le succès de DL appartient à la disponibilité de plus de données d'entrainement. Google, Facebook et Amazon a déjà commencé à l'utiliser pour faire l'analyse de leurs énormes quantités de données [10] [11].

2.2 Les applications du Deep Learning

2.2.1. La reconnaissance faciale : Les yeux, le nez, la bouche, tout autant de caractéristiques qu'un algorithme de DL va apprendre à détecter sur une photo. Il va s'agir en premier lieu de donner un certain nombre d'images à l'algorithme, puis à force d'entraînement, l'algorithme va être en mesure de détecter un visage sur une image.

2.2.2. Le traitement automatique de langage naturel : Le traitement automatique de langage naturel est une autre application du DL. Son but étant d'extraire le sens des mots, voire des phrases pour faire de l'analyse de sentiments. L'algorithme va par exemple comprendre ce qui est dit dans un avis Google, ou va communiquer avec des personnes via des chatbots. La lecture et l'analyse automatique de textes est aussi un des champs d'application du DL avec le Topic Modeling : tel texte aborde tel sujet.

2.2.3. Voitures autonomes : Les entreprises qui construisent de tels types de services d'aide à la conduite, ainsi que des voitures autonomes telles que Google, doivent apprendre à un ordinateur à maîtriser certaines parties essentielles de la conduite à l'aide de systèmes de capteurs numériques au lieu de l'esprit humain. Pour ce faire, les entreprises commencent généralement par entraîner des algorithmes utilisant une grande quantité de données. Vous pouvez imaginer comment un enfant apprend grâce à des expériences constantes et à la réplication. Ces nouveaux services pourraient fournir des modèles commerciaux inattendus aux entreprises.

2.2.4. Recherche vocale et assistants à commande vocale : L'un des domaines d'utilisation les plus populaires de DL est la recherche vocale et les assistants intelligents à commande vocale. Avec les grands géants de la technologie ont déjà fait d'importants investissements dans ce domaine, des assistants à commande vocale peuvent être trouvés sur presque tous les smartphones. Le Siri d'Apple est sur le marché depuis octobre 2011. Google aujourd'hui, l'assistant à commande vocale pour Androïde, a été lancé moins d'une année après Siri. Le plus récent des assistants intelligents à commande vocale est Microsoft Cortana.

2.2.5. Ajout automatique de sons à des films muets : Dans cette tâche, le système doit synthétiser des sons pour correspondre à une vidéo silencieuse. Le système est formé à l'aide de 1 000 exemples de vidéos avec le son d'une baguette frappant différentes surfaces et créant différents sons. Un modèle DL associe les images vidéo à une base de données de sons pré-enregistrés afin de sélectionner le son à reproduire qui correspond le mieux à ce qui se passe dans la scène. Le système a ensuite été évalué à l'aide d'un test de contrôle, comme une configuration dans laquelle les humains devaient déterminer quelle vidéo comportait le son réel ou le son factice (synthétisé).). Ceci utilise à la fois les réseaux de neurones convolutionnels et les réseaux de neurones récurrents à mémoire à court terme.

2.2.6. Traduction automatique : Il s'agit d'une tâche dans laquelle des mots, expressions ou phrases donnés dans une langue sont automatiquement traduits dans une autre langue. La traduction automatique existe depuis longtemps, mais DL permet d'obtenir les meilleurs résultats dans deux domaines spécifiques :

• Traduction automatique de texte
• Traduction automatique d'images

La traduction de texte peut être effectuée sans aucun traitement préalable de la séquence, ce qui permet à l'algorithme d'apprendre les dépendances entre les mots et leur correspondance avec une nouvelle langue.

2.2.7. Génération automatique de texte : C'est une tâche intéressante, où un corpus de texte est appris et à partir de ce modèle, un nouveau texte est généré, mot par mot ou caractère par caractère. Le modèle est capable d'apprendre comment épeler, ponctuer, former des phrases et même capturer le style du texte dans le corpus. Les grands réseaux de neurones récurrents sont utilisés pour apprendre la relation entre les éléments dans les séquences de chaînes d'entrée, puis pour générer du texte.

2.2.8. Reconnaissance d'image : Un autre domaine populaire en matière de DL est la reconnaissance d'image. Son objectif est de reconnaître et d'identifier les personnes et les objets dans les images, ainsi que de comprendre le contenu et le contexte. La reconnaissance d'image est déjà utilisée dans plusieurs secteurs tels que les jeux, les médias sociaux, la vente au détail, le tourisme, etc. Cette tâche nécessite la classification des objets d'une photo parmi un ensemble d'objets connus auparavant. Une variante plus complexe de cette tâche, appelée détection d'objet, consiste à identifier spécifiquement un ou plusieurs objets dans la scène de la photo et à dessiner un cadre autour d'eux.

2.2.9. la description automatique d'image : Le sous-titrage automatique des images est la tâche pour laquelle le système doit générer une légende décrivant le contenu de l'image. Une fois que vous pouvez détecter des objets sur des photographies et générer des étiquettes pour ces objets, vous pouvez voir que l'étape suivante consiste à transformer ces étiquettes en description de phrase cohérente. Généralement, les systèmes impliquent l'utilisation de très grands réseaux de neurones convolutifs pour la détection d'objets sur les photographies, puis d'un réseau de neurones récurrent comme une mémoire à court terme à long terme pour transformer les étiquettes en une phrase cohérente..

2.2.10. Colorisation automatique : La colorisation de l'image pose le problème de l'ajout de couleurs aux photographies noir et blanc. Le DL peut être utilisé pour utiliser les objets et leur contexte dans la photographie pour colorer l'image, un peu comme un opérateur humain pourrait aborder le problème. Cette capacité tire parti des réseaux de neurones de convolution de grande qualité et de très grande taille formés pour ImageNet et cooptés pour le problème de la colorisation de l'image. Généralement, l'approche implique l'utilisation de très grands réseaux de neurones convolutifs et de couches supervisées qui recréent l'image avec l'ajout de couleurs.

2.2.11. la détection du cancer du cerveau : Une équipe de chercheurs français a noté qu'il était difficile de détecter les cellules cancéreuses du cerveau invasives au cours d'une intervention chirurgicale, en partie à cause des effets de l'éclairage dans les salles d'opération. Ils ont découvert que l'utilisation de réseaux de neurones conjointement avec la spectroscopie Raman pendant les opérations leur permettait de détecter les cellules cancéreuses plus facilement et de réduire le cancer résiduel après l'opération.

2.2.12. Analyse des sentiments du texte : De nombreuses applications ont des commentaires ou des systèmes de révision basés sur des commentaires intégrés à leurs applications. La recherche sur le traitement du langage naturel et les réseaux de neurones récurrents ont parcouru un long chemin et il est maintenant tout à fait possible de déployer ces modèles sur le texte de votre application pour extraire des informations de niveau supérieur. Cela peut être très utile pour évaluer la polarité sentimentale dans les sections de commentaires ou pour extraire des sujets significatifs à l'aide de modèles de reconnaissance d'entités nommées.

2.2.13. Recherche en marketing : En plus de rechercher de nouvelles fonctionnalités susceptibles d'améliorer votre application, DL peut également être utile en arrière-plan. La segmentation du marché, l'analyse des campagnes marketing et bien d'autres peuvent être améliorés à l'aide de modèles de régression et de classification DL. Cela vous aidera vraiment beaucoup si vous avez une grande quantité de données. Sinon, vous ferez probablement mieux d'utiliser des algorithmes traditionnels d'apprentissage automatique pour ces tâches plutôt que DL. [12]

2.3 Le réseau neuronal

La pratique, de tous les algorithmes de DL sont des réseaux neuronaux [9]. Les réseaux neuronaux, aussi appelés ANN, sont des modèles de traitement de l'information qui simulent le fonctionnement d'un système nerveux biologique. C'est similaire à la façon dont le cerveau manipule l'information au niveau du fonctionnement. Tous les réseaux neuronaux sont constitués de neurones inter connectés qui sont organisés en couches [9].

2.3.1 Le neurone : Ce qui forme les réseaux de neurones, ce sont les neurones artificiels inspirés du vrai neurone qui existe dans notre cerveau. Les 2 figures suivantes montrent une représentation d'un neurone réel et d'un neurone artificiel :

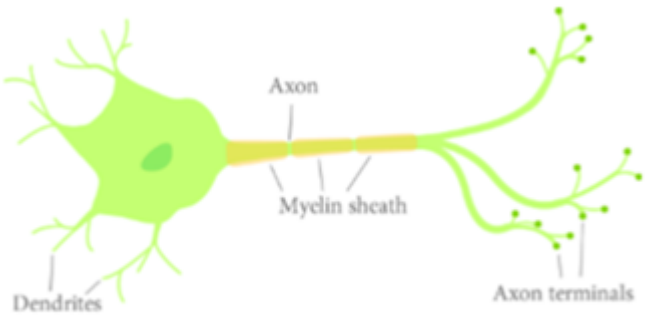


Figure 2.1 – Un neurone réel

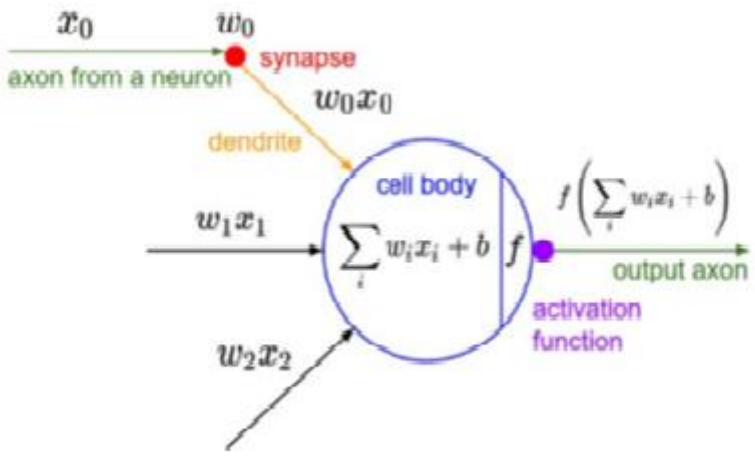


Figure 2.2 – Un neurone artificiel

Les Xi sont des valeurs numériques qui représentent soit les données d'entrée, soit les valeurs sorties d'autres neurones. Les poids Wi sont des valeurs numériques qui représentent soit la valeur de puissance des entrées, soit la valeur de puissance des connexions entre les neurones. Il existe des opérations qui se passent au niveau du neurone artificiel. Le neurone artificiel fera un produit entre le poids (w) et la valeur d'entrée (x), puis ajouter un biais (b), le résultat est transmis à une fonction d'activation (f) qui ajoutera une certaine non-linéarité.

2.3.2 Les fonctions d'activation : La fonction d'activation permet de transformer le signal entrant en signal de sortie. Parmi ces fonctions on a : la tangente hyperbolique, logistique sigmoïde, exponentielle et identité. Le choix de la fonction d'activation dépend de l'application, il a un effet sur l'apprentissage de la représentation et les performances du réseau. Les trois fonctions les plus utilisées sont les fonctions : seuil, linéaire et sigmoïde.

2.3.2.1. Seuil : comme son nom l'indique, cette fonction applique un seuil sur son entrée. Plus précisément, une entrée négative ne passe pas le seuil, la fonction retourne la valeur 0 (faux), alors qu'une entrée positive ou nulle dépasse le seuil, et la fonction retourne 1 (vrai). Il est évidant que ce genre de fonction permet de prendre des décisions binaires.

Nom de la fonction	Relation d'entrée/sortie	Icône
seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$	
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$	
linéaire	$a = n$	
linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$	
linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$	
linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$	
sigmoïde	$a = \frac{1}{1+\exp^{-n}}$	
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
compétitive	$a = 1$ si n maximum $a = 0$ autrement	C

Figure 2.3 – Fonctions d'activation

2.3.2.2. Linéaire : elle affecte directement son entrée à sa sortie selon la relation

$$a = f(n) = n \qquad (2.1)$$

2.3.2.3. Sigmoïde : elle est définit par la relation mathématique :

$$a = \frac{1}{1 + e-n} \qquad (2.2)$$

2.3.3 Les architectures des réseaux de neurones : La plupart des architectures profondes sont réalisées en combinant et recombinant un ensemble limité de primitives architecturales (des couches de réseaux neuronaux) [13]. Selon Hinton, dans [14], nous disons que notre réseau de neurones est profond lorsque le nombre des couches cachées est supérieur à 1. Dans cette section, nous présenterons un bref aperçu des structures communes que l'on retrouve dans de nombreux réseaux profonds.

2.3.3.1 Les réseaux entièrement connectés : Un réseau entièrement connecté (Fully Connected en Anglais) permet de transformer une liste d'entrées en une liste de sorties. La transformation est appelée totalement connectée car toute valeur d'entrée peut affecter toute valeur de sortie. Ces couches auront de nombreux paramètres apprenables, même pour des entrées relativement petites [13], mais elles ont le grand avantage de n'assumer aucune structure dans les entrées. Les calculs sont une série de transformations qui changer les similarités entre les cas. Dans ce type de réseaux, les activités des neurones de chaque couche sont une fonction non-linéaire des activités de la couche inférieure [15].

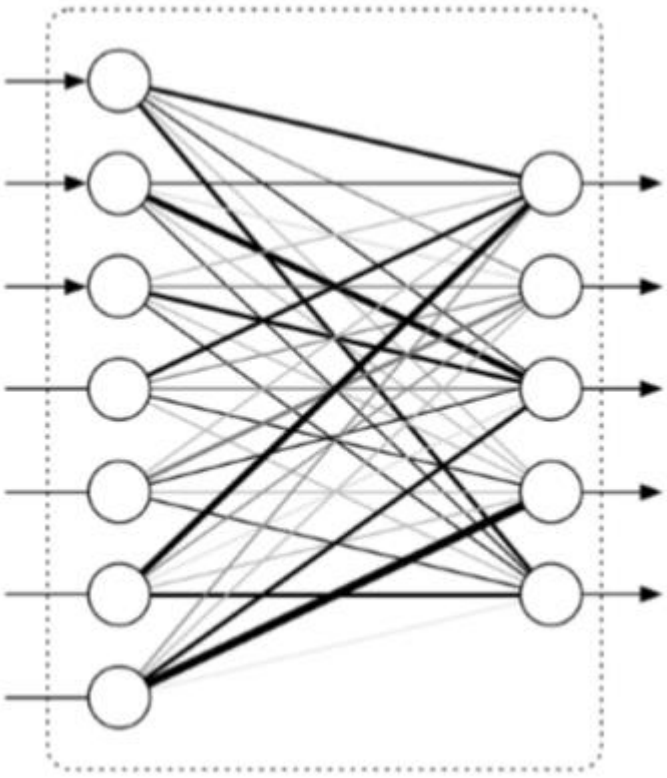


Figure 2.4 – Un réseau entièrement connecté [13].

2.3.3.2. Les réseaux convolutionnels : Un réseau convolutionnel (CNN : Convolutional Neural Networks) suppose une structure spatiale particulière dans son entrée. En particulier, il suppose que les entrées qui sont proches les unes des autres dans l'entrée originale sont sémantiquement liées [13]. CNN est une séquence de couches, et chaque couche transforme un volume d'activations en un autre par une fonction différentiable [16]. Les trois principaux types de couches pour construire ce type de réseau sont : couche convolutive, couche de pooling et couche entièrement connectée [16].

2.3.3.2.1. La couche convolutive : C'est la couche la plus importante et le cœur des éléments constitutifs du réseau convolutif, et c'est aussi elle qui effectue le plus de calculs lourd.

2.3.3.2.2. La couche de pooling : Il est courant d'insérer périodiquement une couche Pooling dans ce type d'architecture. Sa fonction est de réduire progressivement la taille spatiale de la représentation pour réduire le nombre de paramètres et de calculs dans le réseau, et donc de contrôler également le overfitting.

2.3.3.2.3. La couche entièrement connectée : Comme nous l'avons mentionné précédemment, les neurones d'une couche entièrement connectée ont des connexions complètes à toutes les activations de la couche précédente.

2.3.3.3. Les réseaux neuronaux récurrents et LSTM : Les couches de réseaux neuronaux récurrentes (RNN : RecurrentNeuralNetworks) sont des entités primitives qui permettent aux réseaux neuronaux d'apprendre à partir de séquences d'entrées [13].

La figure représente les types de séquences d'entrée que le réseau traite :

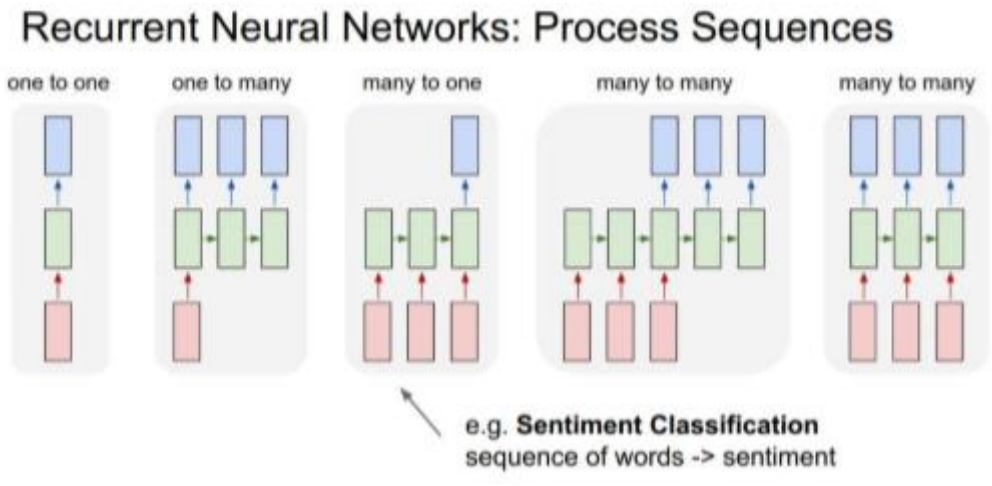


Figure 2.5 – Les types de séquences d'entrée pour un réseau récurrent [13].

Si la séquence que nous traitons est une phrase de 3 termes par exemple, le réseau va être déroulé en un réseau neuronal à 3 couches, une couche pour chaque mot, la figure suivante [13] représente cette idée :

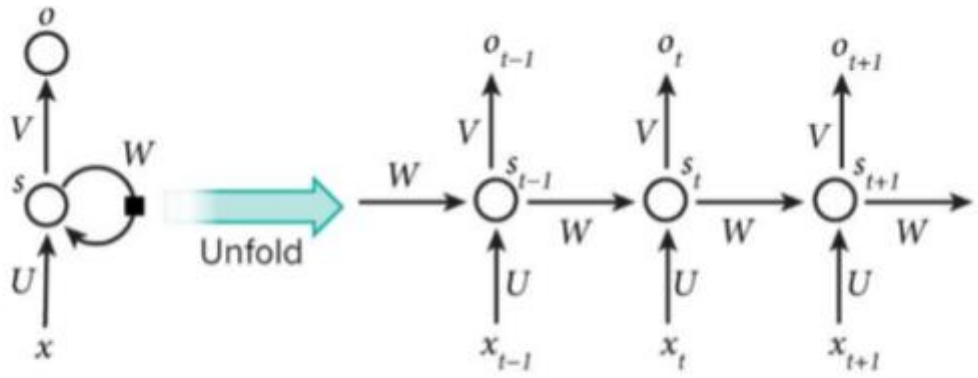


Figure 2.6 – Un exemple d'un réseau récurrent qui se déroule [13].

Les formules qui dirigent le calcul sont les suivantes :

- x_t est l'entrée au temps t
- s_t est l'état caché à l'instant t

s_t est calculé en fonction de l'état caché précédent et de l'entrée à l'étape courante avec : $x_t = f(u_{xt} + w_{st-1})$

- La fonction f est la fonction d'activation
- O_t à l'instant. Par exemple, si nous souhaitons prédire le mot suivant dans une phrase, ce serait un vecteur de probabilités à travers notre vocabulaire [17], nous utiliserons alors la fonction d'activation softmax avec : o_t = softmax(v_{st}) .

Long Short-Term Memory (LSTM) : Les réseaux de mémoire à long terme à court terme généralement appelés simplement (LSTM : Long Short Term Memory) sont un type spécial de RNN. Ils ont été introduits par Hochreiter Schmidhuber (1997). Les Réseaux neuronaux récurrents présentés dans la section précédente sont capables d'apprendre des règles de mise à jour de séquence arbitraire en théorie. Dans la pratique, cependant, ces modèles oublient généralement rapidement le passé [13]. C'est ce qu'on appelle le problème de la disparition de gradient [18] et c'est pourquoi ils ont inventé le LSTM. La cellule LSTM est une adaptation de la couche récurrente qui permet aux signaux plus anciens des couches profondes de se déplacer vers la cellule du présent [13].

La figure suivante représente une chaîne de trois cellules LSTM :

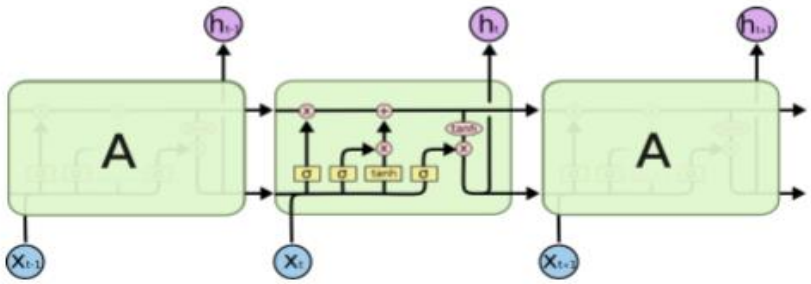
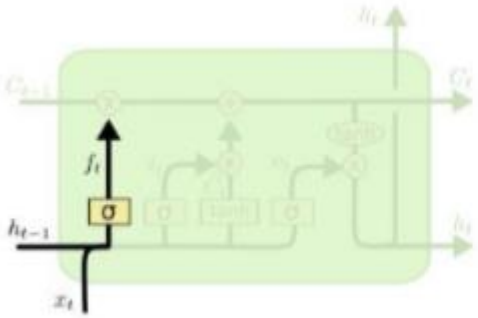


Figure 2.7– Une chaîne de cellules LSTM [19]

Les calculs se déroulent comme suit [19] :



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 2.8– Une cellules LSTM [19]

Avec :

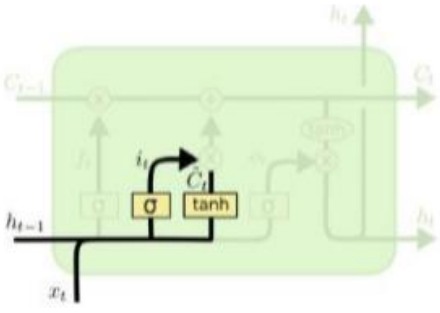
$ht - 1$: La sortie a l'instant t-1

xt : L'entrée courant à l'instant t

b : C'est le bais

w : C'est le poids

σ : C'est la fonction sigmoïde



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

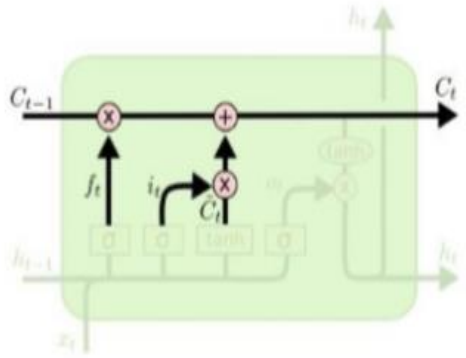
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 2.9– Une cellules LSTM [19]

Avec :

tanh : C'est la fonction d'activation tangente hyperbolique

c_t : Une valeur candidate

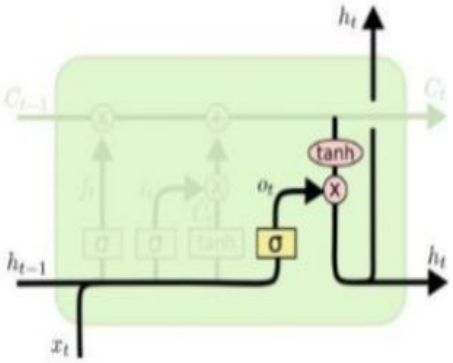


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 2.10– Une cellules LSTM [19]

Avec :

c_t : État interne



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 2.11– Une cellules LSTM [19]

Avec :

h_t : La sortie

Les réseaux neuronaux mentionnés, ne sont pas tous les réseaux qui existent, une excellente ressource qui résume peut-être toute les architectures est en [20]. La figure suivante montre une représentation de ce travail :

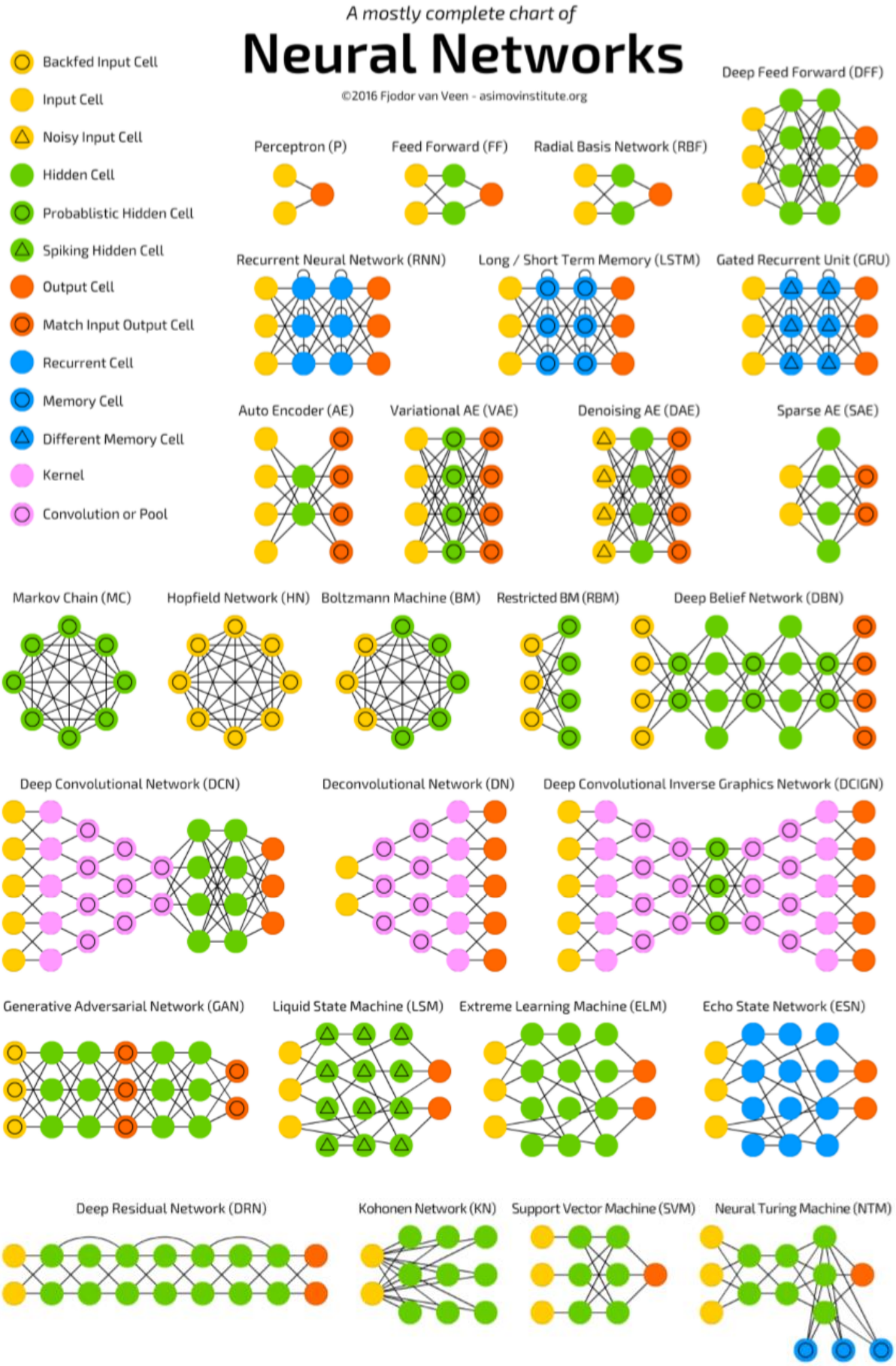


Figure 2.12 – Un résumé des types d'architectures de réseaux de neurones [20].

2.4 L'apprentissage en Deep Learning :

2.4.1 Introduction :

Le processus d'apprentissage dans le DL revient à l'entraînement du réseau neuronal en utilisant des optimiseurs itératifs, qui ne font que conduire la fonction de coût à une très faible valeur [21]. Nous pouvons utiliser différents algorithmes pour effectuer l'apprentissage [9], mais l'algorithme le plus utilisé est l'algorithme itératif d'optimisation par la descente de gradient qu'est la méthode la plus utilisée presque sur tous les réseaux neuronaux avec ses différents modèles [21]. Le processus d'apprentissage revient au problème d'optimisation où il s'agit de minimiser ou de maximiser une fonction f(x). Cette fonction est appelée la fonction objective, les auteurs dans [64] l'ont appelée aussi la fonction de coût, la fonction de perte et la fonction d'erreur. Pendant l'entrainement, l'algorithme tente d'identifier le minimum global sans tomber dans le piège du minimum local. Le schéma suivant réalisé par les auteurs en [64] est une illustration de ce concept :

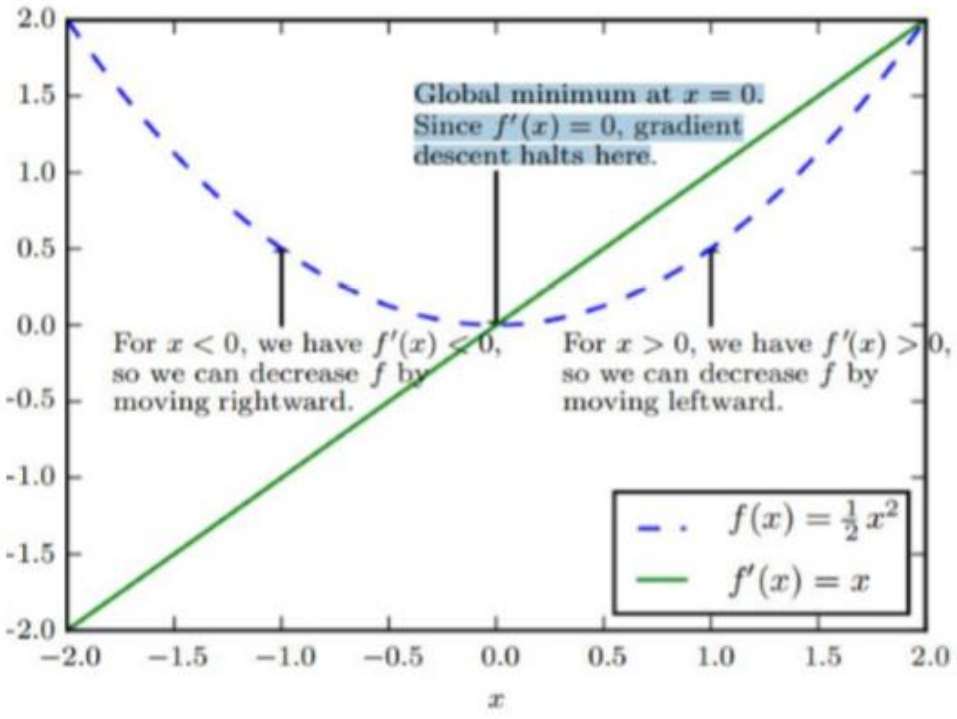


Figure 2.13 – Une illustration du processus de recherche de l'optimum [21].

2.4.2 Les variantes de la descente de gradient :

Il existe trois principaux types de variantes de l'algorithme de descente de gradient. La principale différence entre eux est la quantité de données que nous utilisons lorsque nous calculons le gradient pour chaque étape d'apprentissage [9] [21] [22]. Les algorithmes d'optimisation qui utilisent tout l'ensemble de l'apprentissage sont appelés les méthodes de gradients déterministes ou batch descente de gradient, car ce sont des méthodes où tous les exemples d'apprentissage sont traités simultanément dans un grand batch, tandis que le terme "batch" pour décrire un groupe d'exemples. Les algorithmes d'optimisation qui n'utilisent qu'un seul exemple à la fois sont parfois appelés méthodes stochastiques ou en online descente de gradient. La plupart des algorithmes utilisés pour l'apprentissage profond

se situent quelque part entre les deux, utilisant plus d'un mais moins que tous les exemples d'entrainement. Ils sont appelés les méthodes stochastiques de minibatch ou de minibatch descente de gradient.

2.4.3 Les Algorithmes d'optimisation de la descente de gradient Adam :

Adam est un algorithme d'optimisation présenté en 2015 [80]. Le nom de cet algorithme est dérivé d'Adaptive Moment Estimation. Lors de l'introduction de cet algorithme, les auteurs ont présenté les avantages de l'utilisation d'Adam sur des problèmes d'optimisation non convexes, comme suit :

- Simplicité de mise en œuvre.
- Efficacité du calcul.
- Peu de mémoire requise.
- Bien adapté aux problèmes volumineux en termes de données et/ou de paramètres.
- Les hyper-paramètres nécessitent généralement peu de réglages.

Il existe d'autres optimiseurs avec différents mécanismes de fonctionnement, comme :

- Adagrad
- RMSProp
- Adadelt

2.5 L'apprentissage profond et la détection des spams :

Le spam a commencé à être pris en compte au milieu des années 1990 et il a pris de l'importance avec l'extension d'Internet. cependant, l'étude du spam au sein de la communauté académique est tout à fait récente. La taxonomie des spams a été suggérée par (Gyongyi et al., 2004), la plupart des recherches se concentrant sur certains des principaux types de spam Web, comme par exemple, le contenu, le cloaking et le lien.

En général, les auteurs considèrent la détection de spam comme étant un problème de classification binaire, dont les deux classes sont spam et non-spam, et qui met en jeu plusieurs techniques d'apprentissage (Najork, 2009). Un travail d'investigation met en relief que les travaux se concentrent sur le rôle des techniques d'apprentissage machine dans la détection de spam, sur la façon de préparer et de traiter les données, mais aussi, d'extraire les caractéristiques pertinentes. Généralement, les travaux se concentrent sur la proposition de nouvelles techniques d'apprentissage, ou sur l'exploration de nouveaux attributs pertinents. Ainsi, plusieurs travaux de recherche se penchent sur cette dernière tendance.

Dans notre travail en nous basant sur la façon avec laquelle les données sont préparées. ce travail met en jeu les méthodes d'apprentissage automatique en se basant sur le contenu des messages.

2.6 Conclusion :

Le DL a prouvé son efficacité dans de nombreux problèmes complexes avec l'utilisation de réseaux de neurones artificiels pour apprendre et extraire des modèles et des informations significatives depuis les données. Par conséquent, nous trouvons de nombreuses contributions qui tentent d'adapter cette approche comme une solution au problème de détection de spans.

Chapitre 3 – Conception et implémentation

3.1 Introduction :

Ce chapitre présente quelques techniques de la détection de spam basées sur l'apprentissage profond.

3.2 Outils utilisés :

Dans ce qui suit, nous présentons les outils utilisés lors de l'implémentation.

3.2.1 Configuration utilisée :

La configuration du matériel utilisé dans notre implémentation est :

- Un PC portable DELL Core i5 CPU 2.60 GHZ.
- Une RAM de taille 4 GO.
- Un disque dur de taille 500 GO.
- Un système d'exploitation Windows 10 (64 bit)

3.2.2 Description du corpus utilisé : Le corpus SMS Spam Collection v.1 [24] est un ensemble commun de messages étiquetés SMS. Il dispose d'une collection composée de 5 574 messages en anglais, réels, étiqueté selon étant légitime (Ham) ou spam. Cette collection contient 747 messages spam et 4827 messages légitimes.

3.2.2.1. Utilisation : La collection est composée d'un seul fichier texte où chaque ligne contient la bonne classe suivie par le message brut. Nous vous proposons quelques exemples ci-dessous:

```
spam    Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA
to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
ham     U dun say so early hor... U c already then say...
ham     Nah I don't think he goes to usf, he lives around here though
spam    FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like
some fun you up for it still? Tb ok! XxX std chgs to send, £1.50 to rcv
ham     Even my brother is not like to speak with me. They treat me like aids
patent.
ham     As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has
been set as your callertune for all Callers. Press *9 to copy your friends
Callertune
spam    WINNER!! As a valued network customer you have been selected to receivea
£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
spam    Had your mobile 11 months or more? U R entitled to Update to the latest
colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030
ham     I'm gonna be home soon and i don't want to talk about this stuff anymore
tonight, k? I've cried enough today.
spam    SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to
87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info
spam    URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot!
Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18
```

3.2.3 Langage de programmation et librairies :

3.2.3.1. Python :

Un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Il est développé depuis 1989 par Guido van Rossum et comme la plupart des applications et outils open source, maintenu par une équipe de développeurs un peu partout dans le monde. Le lien du site officiel1. Parmis les caractéristiques de python et qui nous ont pousser l'utiliser on trouve : (van Rossum, 2009)

- Python est gratuit, mais on peut l'utiliser sans restriction dans des projets commerciaux.
- Python intègre, comme Java ou les versions récentes de C++, un système d'exceptions, qui permettent de simplifier considérablement la gestion des erreurs.
- Python est orienté-objet. Il supporte l'héritage multiple et la surcharge des opérateurs. Dans son modèle objets, et en reprenant la terminologie de C++, toutes les méthodes sont virtuelles.
- Python est extensible et dynamique
- Python possède actuellement deux implémentations. L'une, interprétée, dans laquelle les programmes Python sont compilés en instructions portables, puis exécutés par une machine virtuelle (comme pour Java, avec une différence importante : Java étant statiquement typé, il est beaucoup plus facile d'accélérer l'exécution d'un programme Java que d'un programme Python). L'autre génère directement du bytecode Java. [25]

3.2.3.2. Tensorflow :

Une bibliothèque de logiciels libres et open-source pour le flux de données et la programmation différentiable a` travers un éventail de tâches. Il s'agit d'une bibliothèque mathématique symbolique, et elle est également utilisée pour des applications d'apprentissage machine telles que les réseaux neuronaux. Elle est utilisée à la fois pour la recherche et la production chez Google. TensorFlow a été développé par l'équipe Google Brain pour une utilisation interne à Google. Il a été publié sous la licence Apache 2.0 le 9 novembre 2015.

3.2.3.3. Keras :

Une API de réseaux neuronaux de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow, CNTK ou Theano. Il a ´été développé dans le but de permettre une expérimentation rapide. Pouvoir passer de l'idée au résultat avec le moins de retard possible est la clé d'une bonne recherche. Ses caractéristiques sont :

- Permet un prototypage facile et rapide (grâce à la convivialité, la modularité et l'extensibilité).
- Supporte à la fois les réseaux convolutifs et les réseaux récurrents, ainsi que les combinaisons des deux.
- Fonctionne de manière transparente sur CPU et GPU.

3.2.3.4. Scikit-learn :

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

3.2.3.5. Pandas :

Est une bibliothèque écrite pour le langage de programmation 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 de séries temporelles.

Fonctionnalités de la bibliothèque :

- l'objet DataFrame pour manipuler des données aisément et efficacement avec des index pouvant être des chaines de caractères ;
- des outils pour lire et écrire des données structurées en mémoire depuis et vers différents formats : fichiers CSV, fichiers textuels, fichier du tableur Microsoft Excel, base de données SQL ou le format rapide et permettant de gérer de gros volume de données nommé HDF5.
- alignement intelligent des données et gestion des données manquantes (NaN = not a number). alignement des données basé sur des étiquettes (chaines de caractères). tri selon divers critères de données totalement désordonnées.
- Redimensionnement et table pivot ou pivot table en anglais (aussi nommé tableau croisé dynamique) .
- Fusion et jointure de large volume de données.
- Analyse de séries temporelles.

3.2.3.6. NumPy :

Est une bibliothèque python utilisée pour travailler avec des tableaux. Il a également des fonctions pour travailler dans le domaine de l'algèbre linéaire, de la transformée de Fourier et des matrices. NumPy a été créé en 2005 par Travis Oliphant. C'est un projet open source et vous pouvez l'utiliser librement. NumPy signifie Numerical Python.

Pourquoi utiliser NumPy : En Python, nous avons des listes qui servent à des tableaux, mais elles sont lentes à traiter. NumPy vise à fournir un objet tableau jusqu'à 50 fois plus rapide que les listes Python traditionnelles. L'objet tableau dans NumPy s'appelle ndarray, il fournit de nombreuses fonctions de support qui facilitent le travail avec ndarray. Les tableaux sont très fréquemment utilisés dans la science des données, où la vitesse et les ressources sont très importantes.

3.3 Architectures proposées :

Au cours de nos expériences, nous avons créé deux modèles avec des architectures déférentes, où le premier était un modèle CNN (Convolutional Neural Networks) et le second était un modèle RNN (recurrent neural network)

3.3.1 Architecture 01 :

Le premier modèle que nous présentons est constitué de couche embedding et couche de convolution suivies de couche de dropout et couche pooling (Global_Maxpooling) et deux couches dropout et deux couches fully connected. Le message en entrée passe d'abord à la couche de convolution ont un ensemble de 16 filtres de taille 3*3 et une fonction d'activation Relu. Chacune des couches est suivie d'une couche Pooling avec des fenêtres de taille 2*2. à la sortie de la couche couche Pooling, nous aurons 32 feature maps de taille 10*10. Le vecteur de caractéristiques issu des convolutions a donc une dimension de 3200. La première couche fully connected calcule un vecteur de taille 32 et suivie d'une couche

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 20, 10)            50010
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 16, 16)            816
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 16)            0
_________________________________________________________________
global_max_pooling1d_1 (Glob (None, 16)                0
_________________________________________________________________
dropout_2 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 136
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 9
=================================================================
Total params: 50,971
Trainable params: 50,971
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 3.1 – Représentation synthétique de l'architecture 01

Relu et un dropout qui est égal à 0,2. La deuxième couche fully connected renvoie un vecteur de probabilités de taille 5 (le nombre de classes) et la fonction softmax lui est appliquée.

3.3.2 Architecture 02 :

Le deuxième modèle se compose de 3 couches : couche Embedding et une couche SimpleRNN et une couche fully connected. La première couches de embedding ont un ensemble de 32 filtres de taille 3*3 et une fonction d'activation Relu. Les couches de simpleRNN ont un ensemble de 32 filtres de taille 3*3 et la fonction d'activation Relu est utilisée dans les deux couches. Elles sont aussi suivies d'une couche fully connected avec des fenêtres de taille 2*2. La seule différence avec les couches 1 et 2 est que l'ensemble de filtres appliqué est de taille 64. 64 feature maps de taille 2*2.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 32)          320000
_________________________________________________________________
simple_rnn_1 (SimpleRNN)     (None, 32)                2080
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 33
=================================================================
Total params: 322,113
Trainable params: 322,113
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 3.2 – Représentation de model de l'architecture 02

Pour la dernière couche, la couche fully connected, la fonction softmax est appliquée, et le vecteur de probabilités renvoyé est de taille 6 (le nombre de classes).

3.4 Résultats et discussions :

Pour avoir des résultats satisfaisants un nombre de paramètres doivent être bien choisis :

- Taille du dropout : le dropout est une technique simple et puissante de régularisation pour les réseaux de neurones et les modèles d'apprentissage profond. Le dropout consiste ignorer les neurones sélectionnes au hasard pendant la phase d'entrainement. Ils sont abandonnés au hasard. Cela signifie que leur contribution à l'activation des neurones en aval est temporairement supprimée lors du passage en avant et que les mises à jour de poids ne sont pas appliquées au neurone lors du passage en arrière. En générale, utilisez une petite valeur de dropout de 20% a` 50% des neurones, 20% constitue un bon point de départ. Une probabilité trop faible a un effet minimal et une valeur trop élevée entraîne un sous-apprentissage du réseau. (Brownlee, 2016)
- Flatten : son rôle est de convertir un vecteur multidimensionnel en un vecteur à une dimension i.e; Supposons que nous utilisons un CNN dont les couches initiales sont des couches de convolution et de pooling. Les couches ont des vecteurs multidimensionnels comme sorties. Pour utiliser une couche dense (une couche

entièrement connectée) après les couches de convolution, nous devons «désempiler»
tout le vecteur multidimensionnel en un très long vecteur 1D et ce grâce à la couche flatten.(Zia, 2018)

- Taille du batch size : elle définit le nombre d´échantillons qui seront propagés sur le réseau. Par exemple, supposons que nous avons 1050 échantillons d'apprentissage et que nous souhaitons configurer un batch size égal à 100. L'algorithme extrait les 100 premiers échantillons (de a à 100) de l'ensemble de données d'apprentissage et forme le réseau. Ensuite, il prélève les 100 derniers échantillons (de 101 a` 200) et entraîne à nouveau le réseau. Nous pouvons continuer à suivre cette procédure jusqu'a` ce que tous les échantillons soient propagés sur le réseau. (CrossValidated, 2015)
- Nombre de steps per epochs : c'est le nombre total d'étapes (lots d'échantillons) avant de déclarer une itération terminée et de commencer la suivante. Lors de l'entrainement avec des vecteurs d'entrée tels que les vecteurs de données TensorFlow, la valeur par défaut Aucune est égale au nombre d´echantillons du jeu de données divisé par la taille du lot, ou 1 si cela ne peut pas être déterminé. (Keras, 2016)
- Nombre d'épochs : est une limite arbitraire, généralement définie comme un passage sur l'ensemble de données complet, utilisée pour séparer la formation en phases distinctes, ce qui est utile pour la journalisation et l'évaluation périodique (Keras, 2016). En d'autres termes, le nombre d'époques signifie combien de fois vous passez par votre donnée d'entrainement.

Pour montrer les résultats obtenus des trois modèles, nous présentons dans ce qui suit les résultats en termes de précision, d'erreur, matrice de confusion et rapport de classification pour chacun d'entre eux.

3.4.1 Résultats du premier modèle :

Les figures 3.3 et 3.4 présentent la précision et l'erreur du modèle selon l'apprentissage et le test. Nous remarquons que la précision de l'apprentissage et du test augmente avec le nombre d'épochs, ce qui signifie qu'à chaque epochs (itération) le modèle apprend plus d'informations, tendis que l'erreur d'apprentissage et de la validation diminue avec le nombre d'épochs.

Une matrice de confusion est un tableau qui est souvent utilisé pour d´écrire la performance d'un modèle de classification sur un ensemble de données de test dont les valeurs réelles sont connues (DATASCHOOL, 2014). Elle refléte les métriques du vrai positif, vrais négatif, faux positifs et faux négatifs.

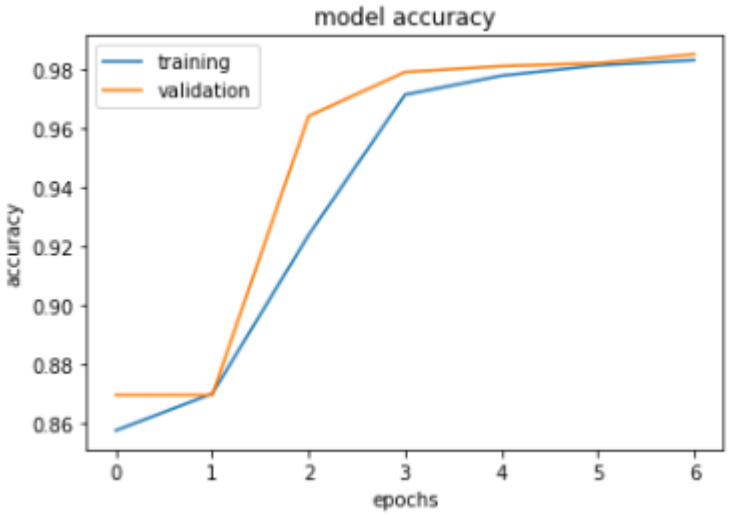


Figure 3.3 – Précision du modèle 01

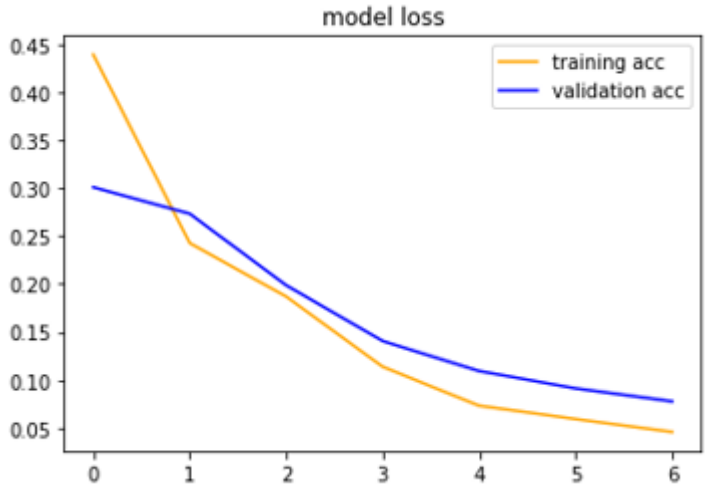


Figure 3.4 – Erreur du modèle 01

Vrais positifs (TP) : ce sont des cas dans lesquels le modèle a prédit qu'ils sont positifs et qu'ils le sont en réalité.

Vrais négatifs (TN) : des cas dans lesquels le modèle à prédit qu'ils sont négatifs, et ils le sont en réalité.

Faux positifs (FP) : le modèle les a prédits en tant que positifs, mais ils ne le sont pas.

Faux négatifs (FN) : le modèle a prédit qu'ils sont négatifs, alors qu'ils ne le sont pas.

	Positif	Négatif
Positif	TP	FN
Négatif	FP	TN

Table 3.1 – Modelé d'une matrice de confusion

La Représentation matricielle de la phrase avec 10 dimensions dans la figure 3.2, nous déduisons que parmi un ensemble des messages constitué de `4457` un nombre de `1115` messages ont été bien classées. Le modèle a fait un bon apprentissage des classes des émotions exprimant : spam, ham.

			Dimensions									
			0	1	2	3	4	5	6	7	8	9
Words	yo	0	0.06	0.00	-0.02	0.01	0.04	-0.03	0.01	-0.04	-0.05	0.06
	we	1	0.08	0.01	-0.04	-0.06	-0.07	-0.02	0.00	0.05	-0.07	0.03
	are	2	-0.01	0.00	-0.05	0.01	-0.04	-0.05	0.03	0.04	-0.03	-0.01
	watching	3	0.07	-0.01	-0.08	-0.01	-0.03	-0.01	0.02	0.01	-0.07	0.06
	a	4	0.00	0.02	0.02	0.06	0.03	-0.01	0.09	0.03	-0.01	0.03
	movie	5	0.07	-0.07	-0.06	0.00	-0.04	-0.01	0.04	-0.02	-0.01	0.04
	on	6	-0.07	-0.01	0.04	0.02	0.00	0.03	0.04	-0.03	0.05	-0.04
	netflix	7	0.02	0.04	-0.03	0.04	0.02	0.00	0.05	0.03	-0.01	-0.01

Table 3.2 – Représentation matricielle de la phrase avec 10 dimensions du premier modèle

Le rapport de classification affiche les scores de : précision, rappel, F1-mesure et support pour le modèle. Le tableau 3.3 est le rapport de classification du premier modèle.

Précision : est la capacité d'un classificateur à ne pas étiqueter une instance positive qui est effectivement négative. Pour chaque classe, il est défini comme le rapport entre les vrais positifs et la somme des vrais et des faux positifs. En d'autres termes, "pour tous les cas classés positifs, quel pourcentage était correct?"

$$Précision = \frac{TP}{TP + FP} \qquad (3.1)$$

Rappel : est la capacité d'un classificateur à trouver toutes les instances positives. Pour chaque classe, il est défini comme le rapport entre les vrais positifs et la somme des vrais positifs et des faux négatifs. En d'autres termes, "pour tous les cas positifs, quel pourcentage a été classé correctement?"

$$Rappel = \frac{TP}{TP + FN} \qquad (3.2)$$

F1-mesure : une moyenne harmonique pondérée de précision et de rappel telle que le meilleur score est 1 et le pire est 0. En général, les scores F1 sont inférieurs aux mesures de précision car ils intègrent la précision et le rappel dans leur calcul. En règle générale, la moyenne pondérée de F1 devrait être utilisée pour comparer les modèles de classificateurs, et non la précision globale.

$$F1Mesure = 2 * \left(\frac{Rappel * Précision}{Rappel + Précision}\right) \qquad (3.3)$$

Support : Le support est le nombre d'occurrences réelles de la classe dans l'ensemble de données spécifié. Un soutien déséquilibré dans les données sur la formation peut indiquer des faiblesses structurelles dans les scores déclarés du classificateur et pourrait indiquer la nécessité d'un échantillonnage stratifié ou d'un rééquilibrage. Le support ne change pas entre les modèles mais diagnostique plutôt le processus d'évaluation.

```
Train on 4011 samples, validate on 1003 samples
Epoch 1/7
4011/4011 [==============================] - 2s 495us/step - loss: 0.4390 - accuracy: 0.8574 - val_loss: 0.3014 - val_accuracy: 0.86942s
Epoch 2/7
4011/4011 [==============================] - 1s 174us/step - loss: 0.2429 - accuracy: 0.8699 - val_loss: 0.2735 - val_accuracy: 0.8694
Epoch 3/7
4011/4011 [==============================] - 1s 150us/step - loss: 0.1884 - accuracy: 0.9237 - val_loss: 0.2017 - val_accuracy: 0.9641
Epoch 4/7
4011/4011 [==============================] - 1s 224us/step - loss: 0.1158 - accuracy: 0.9713 - val_loss: 0.1412 - val_accuracy: 0.9791
Epoch 5/7
4011/4011 [==============================] - 1s 176us/step - loss: 0.0752 - accuracy: 0.9778 - val_loss: 0.1099 - val_accuracy: 0.9811
Epoch 6/7
4011/4011 [==============================] - 1s 148us/step - loss: 0.0617 - accuracy: 0.9813 - val_loss: 0.0909 - val_accuracy: 0.9821
Epoch 7/7
4011/4011 [==============================] - 1s 175us/step - loss: 0.0490 - accuracy: 0.9830 - val_loss: 0.0770 - val_accuracy: 0.9850
```

Figure 3.5 – Rapport de validation et training du premier modèle

	Précision	Rappel	F1-mesure	Support
spam	0.9838	0.971	0.968	207
ham	0.931	0.939	0.918	137

Table 3.3 – Rapport de classification du premier modèle

3.4.2. Résultats du deuxième modèle :

Les figures 3.6 et 3.7 présentent la précision et l'erreur du modèle selon l'apprentissage et le test. Comme pour le modèle précédent, nous remarquons que plus le nombre d'époch augmente plus la valeur de la précision de l'apprentissage et du test augmente, tendis que la valeur de l'erreur d'apprentissage et de la validation diminue.

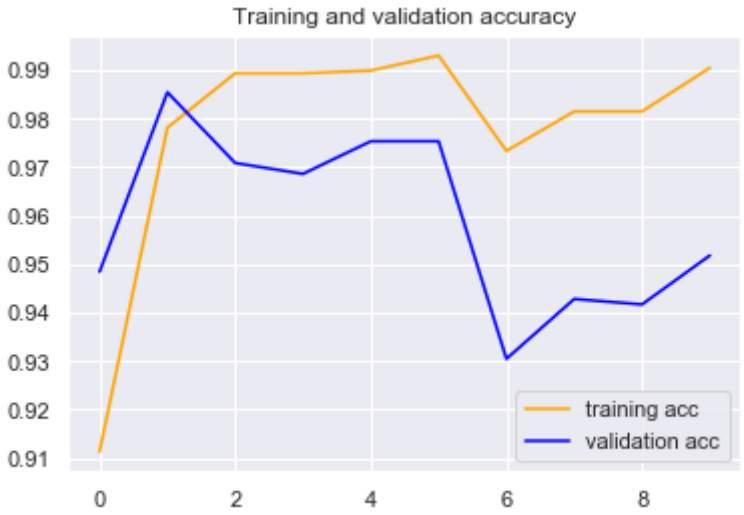


Figure 3.6 – Précision du modèle 02

Figure 3.7 – Erreur du modèle 02

Dans la figure 3.8 et du rapport de validation et training, nous remarquons que les émotions qui ont été bien classées par ce modèle sont : spam, ham.

```
Train on 3565 samples, validate on 892 samples
Epoch 1/10
3565/3565 [==============================] - 17s 5ms/step - loss: 0.2776 - acc: 0.9114 - val_loss: 0.1539 - val_acc: 0.9484
Epoch 2/10
3565/3565 [==============================] - 12s 3ms/step - loss: 0.0848 - acc: 0.9781 - val_loss: 0.0596 - val_acc: 0.9854
Epoch 3/10
3565/3565 [==============================] - 11s 3ms/step - loss: 0.0434 - acc: 0.9893 - val_loss: 0.1011 - val_acc: 0.9709
Epoch 4/10
3565/3565 [==============================] - 12s 3ms/step - loss: 0.0394 - acc: 0.9893 - val_loss: 0.0871 - val_acc: 0.9686
Epoch 5/10
3565/3565 [==============================] - 12s 3ms/step - loss: 0.0353 - acc: 0.9899 - val_loss: 0.0665 - val_acc: 0.9753
Epoch 6/10
3565/3565 [==============================] - 12s 3ms/step - loss: 0.0241 - acc: 0.9930 - val_loss: 0.0724 - val_acc: 0.9753
Epoch 7/10
3565/3565 [==============================] - 12s 3ms/step - loss: 0.0778 - acc: 0.9734 - val_loss: 0.1911 - val_acc: 0.9305
Epoch 8/10
3565/3565 [==============================] - 15s 4ms/step - loss: 0.0603 - acc: 0.9815 - val_loss: 0.1680 - val_acc: 0.9428
Epoch 9/10
3565/3565 [==============================] - 17s 5ms/step - loss: 0.0568 - acc: 0.9815 - val_loss: 0.1499 - val_acc: 0.9417
Epoch 10/10
3565/3565 [==============================] - 13s 4ms/step - loss: 0.0314 - acc: 0.9905 - val_loss: 0.1332 - val_acc: 0.9518
```

Figure 3.8 – Rapport de validation et training du deuxième modèle

	Précision	Rappel	F1-mesure	Support
spam	0.965	0.967	0.964	200
ham	0.936	0.945	0.936	144

Table 3.4– Rapport de classification du deuxième modèle

Le tableau 3.5 résume les résultats obtenus pour les deux modèles proposés. D'après ces résultats, nous remarquons que la performance d'un modèle ne dépend pas forcément du nombre d'époch mais aussi du nombre de couches de utilisé (plus le nombre est élève plus la précision du modèle est satisfaisante) et la taille du batch size utilisée (plus elle est grande plus nous avons un bon modèle d'entrainement). Le temps d'exécution dépend de la complexité du modèle (plus nous avons de couches fully connected, le temps d'exécution prend plus de temps). Donc, nous pouvons constater que le première modèle est le plus fiable parmi les deux modèles proposés puisque nous avons une précision de 98.38%.

	Précision	F1_mesure	Erreur	Batch_size	Epochs
Architecture01	0.9838	0.968	0.0482	32	7
Architecture02	0.965	0.964	0.13	60	10

Table 3.5 – Tableau de comparaison des deux modèles

```
In [23]: # Predict binary and probabilities
         predictions_df = pd.DataFrame(model.predict(testFeatures))
         predictions_binary_df = round(predictions_df)
         accuracy_score(testLabels, predictions_binary_df)

Out[23]: 0.9838709677419355

In [24]: predictions_binary_df[0].value_counts(dropna=False)

Out[24]: 0.0    482
         1.0    76
         Name: 0, dtype: int64
```

Figure 3.9 – Prédiction du premier modèle

```
Entrée [11]: pred = model.predict_classes(texts_test)
             acc = model.evaluate(texts_test, y_test)
             proba_rnn = model.predict_proba(texts_test)
             from sklearn.metrics import confusion_matrix
             print("Test loss is {0:.2f} accuracy is {1:.2f}  ".format(acc[0],acc[1]))
             print(confusion_matrix(pred, y_test))

             1115/1115 [==============================] - 1s 1ms/step
             Test loss is 0.13 accuracy is 0.96
             [[939  23]
              [ 22 131]]
```

Figure 3.10 – Prédiction du deuxième modèle

Pour évaluer nos modèles, nous avons comparé les résultats obtenus avec des études faites dans le même sujet.

(Araujo et Martinez-Romo, 2010) : Nous comparons nos résultats obtenus avec le travail de (Araujo et Martinez-Romo, 2010) pour détecter les messages spams. Ils ont utilisé un système de détection de spam basé sur un classificateur qui combine de nouvelles caractéristiques basées sur le contenu avec des modèles de langage (LM). Ils ont utilisé l'arbre de décision (C4.5), Bayésien Naïf (NB), SVM et Logistic Regression (LR) comme classificateurs pour leur travail expérimental. Les données WebsparnUK 2007 ont été aussi utilisées. Les résultats des classificateurs de spam utilisant différentes caractéristiques sur l'ensemble d'apprentissage, ont donné une F-mesured'environ 0,40 et une AUC d'environ 0,76, ce qui est significativement inférieur à nos résultats.

(Gongwena et al., 2016) : Notre travail a aussi été comparé à (Gongwena et al., 2016). Dans leur travail, un nouvel algorithme de classement des messages a été proposé. Dans cette méthode, le score de classement des messages est calculé par la méthode TrustRank combinant la diversité des messages et la distribution des caractéristiques de contenu des messages. Certaines caractéristiques, comme le nombre de mots de titre et le ratio de compression desmessages, ont été extraites pour les caractéristiques basées sur le contenu. De même, pour la diversité du Contenu des messages, ils ont analysé les informations de lien de message. WebspamUK-2007 a été utilisé à des fins expérimentales. On trouve à partir des résultats de (Gongwena et al. 2016) qu'ils ont atteint une F-mesure d'environ 0,822 avec un taux de précision de 0,926 et de rappel de 0,739, ce qui est aussi inférieur à nos résultats.

(Rajendra et al. 2016) Enfin, nous avons comparé nos résultats à (Rajendra et al. 2016). Ces derniers ont proposé une approche combinant du contenu et des techniques basées sur le Contenu des messages pour identifier les messages spams. Les caractéristiques basées sur le contenu incluent, la densité des termes et le test du rapport des composantes du langage (Part Of Speech). Concernant l'approche basée sur le Contenu des messages, ils ont exploré la détection collaborative à l'aide du classement de la message personnalisée pour classer comme spam ou non-spam. Ils ont utilisé l'ensemble de données WebspamUK-2006 pour leurs expériences et ont comparé leurs résultats à certaines approches existantes. Leur approche surpasse clairement quatre autres travaux (Egele et al. , 2011 ; Dai et al. , 2009; Becchetti et al., 2008b; Benczur et al., 2007) de détection de spam.

Ils ont aussi atteint une F-mesure d'environ 0.752 avec une précision d'environ 0.729 et un rappel de 0.776, ce qui est encore inférieur à nos résultats.

À partir des comparaisons ci-dessus, il apparait que notre méthode surpasse clairement les trois approches de détection de spam citées (Tableau 3.5).

Approches	**F-Mesure**	
Araujo et Martinez-Romo, 2010	0,4	
Gongwena et al. 2016	0,822	
Rajendra et al. 2016	0,752	
Notre approche	modèle 01	0.968
	modèle 02	0.964

Table 3.6 – Comparaison de F1_mesure des architectures.

3.5 Conclusion :

Dans ce chapitre, nous avons vu à travers les expériences qui ont été faites, l'apport et la contribution de nos caractéristiques au monde de la détection de spam. Pour des résultats plus précis et pour prouver leur efficacité, nous avons utilisé deux modèles d'apprentissage différents. L'application de différents modèles au même dataset montre que le nombre d'époch, la profondeur des réseaux et la taille du batch size sont des facteurs importants pour l'obtention de meilleurs résultats. Après des comparaisons faites avec d'autres travaux effectués dans le même cadre. Les résultats ont été par la suite discutés et interprétés.

CONCLUSION GENERALE

Le domaine de détection de spam a particulièrement progressé ces dix dernières années, grâce à l'introduction des techniques héritées de l'apprentissage profond qui ont amélioré significativement le taux de la détection de spam, par la progression de classification des emails et les messages en spam et légitime.

De nombreux algorithmes d'apprentissage peuvent être utilisés pour la détection de spam. Cet article proposait CNN et RNN pour la classification des spams, avec un F1-mesure de 0.968 et une précision de 98,38%, il est prouvé que ce modèle peut mieux fonctionner par rapport à de nombreuses autres techniques de filtrage comme SVM , etc.

Les résultats des différentes expérimentations réalisées nous ont montré que notre approche donnait l'un des meilleurs résultats dans la classification. Ces résultats se traduisent par un grand pourcentage de rappel de courriers indésirables et légitimes.

Dans notre dernière expérimentation, nous avons étudié la manière dont notre approche se comporte par rapport à l'apprentissage profond, les réseaux de neurones (CNN, RNN) se sont révélés être des classifieurs très performants. Il faut noter que cette propriété est avantageuse.

Enfin, nous avons vu que les réseaux neurones (CNN, RNN) n'échappaient pas aux difficultés d'utilisation des méthodes statistiques. Le fait qu'ils aient été présentés, il y a une dizaine d'années, comme un outil "miraculeux" qui supprimerait toutes les difficultés liées à l'utilisation de ces méthodes a conduit à des traitements complètement erronés des problèmes.

Bibliographie

[1] wikipedia, www.wikipedia.com, 14/04/2018

[2] arobase, www.arobase.org, 28/04/2018

[3] aidewindows, www.aidewindows.net/phishing.php, 28/04/2018

[4] sebsauvage, www.sebsauvage.net/comprendre/spam/index.html, 28/04/2018

[5] B. Hassan, Algorithme de boosting et méta-heuristique basée sur la PSO Pour La détection et le Filtrage De Spam, Thèse de Master, Université Tahar Moulay-SAIDA, 2013

[6] statista, www.statista.com, 29/04/2018

[7] S. Gastellier-prevost, Le spam, 2009

[8] Deep Learning Yann LeCun, Yoshua Bengio Geoffrey Hinton. 2015

[9] Python Deep Learning. Ivan Vasilev, Daniel Slater, Gianmario Spacagna, Peter Roelants, Valentino Zocca. 2019

[10] https ://ai.google, 20 Avril 2019

[11] https ://ai.facebook.com/tools, 22 Avril 2019

[12] Top 15 Deep Learning applications that will rule the world in 2018 and beyond, Vartul Mittal, 3 Oct 2017.

[13] TensorFlow for Deep Learning. Reza Bosagh Zadeh, Bharath Ramsundar. 2017

[14] Deep Learning with Keras. Antonio Gulli, Sujit Pal. 2017

[15] https ://www.cs.toronto.edu/tijmen/csc321/slides/lecture slides lec2.pdf . 2014

[16] http ://cs231n.github.io/convolutional-networks/overview 25 Mars 2019.

[17] http ://www.wildml.com/2015/09/recurrent-neural-networks-tutorial -part-1-introduction-to-rnns/

[18] THE VANISHING GRADIENT PROBLEM DURING LEARNING RECURRENT NEURAL NETS AND PROBLEM SOLUTIONS. Sepp Hochreiter. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems

[19] https ://colah.github.io/posts/2015-08-Understanding-LSTMs/

[20] http ://www.asimovinstitute.org/neural-network-zoo/ 18 Mai 2019.

[21] Deep Learning. Ian Goodfellow, Yoshua Bengio, Aaron Courville. MIT Press. 2016

[22] Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms. Nikhil Buduma. 2017

[23] ADAM : A METHOD FOR STOCHASTIC OPTIMIZATION. Diederik P. Kingma, Jimmy Lei Ba. 2017

[24] Department of Telematics, www.dt.fee.unicamp.br/~tiago/smsspamcollection, consulté le: 15/01/2019

[25] https ://www.python.org/

[26] G. Schryen, Anti-Spam Measures Analysis and Design, Berlin Heidel berg New York, Springer, 2010.

[27] anti-spam, www.anti-spam.fr, 02/05/2018

[28] Nouman Azam, Comparative Study of Features Space Reduction Techniques for Spam Detection, Thèse de Master, National University of Sciences & Technology, Pakistan.

[29] Frameip, www.frameip.com/spam-ham-antispam, 04/05/2018