



Université d'ANTANANARIVO
Domaine Sciences et Technologies
Mention Mathématiques et Informatique

Mémoire en vue de l'obtention du diplôme de Master 2 en
Mathématiques Informatique et Statistique Appliquées

**Annotation automatique d'images par apprentissage
profond :
Génération automatique de descriptions d'une image**

Présenté le 16 Décembre 2016 par :
Nomena Fitiavana NY HOAVY

Devant le jury composé de :

Président du jury :	<i>M. Patrick RABARISON</i>	Université d'Antananarivo
Examinateur :	<i>M. Arthur RANDRIANARIVONY</i>	Université d'Antananarivo
Encadrante :	<i>M^{me} Josiane MOTHE</i>	Université de Toulouse
Co-encadrant :	<i>M. Olivier ROBINSON</i>	Université d'Antananarivo

Remerciements

Je tiens à exprimer ma gratitude et mes remerciements à tous ceux qui m'ont, de près ou de loin, apporté leur aide et qui ont contribué à l'élaboration de ce mémoire.

Je tiens à remercier, Messieurs Olivier ROBINSON, Marc Jany RABIAZAMAHOLY, Andry RASOANAIVO, Tahiry ANDRIAMAROZAKANIAINA ainsi que le corps enseignant de la MISA de m'avoir permis de suivre cette formation, pour les connaissances et les conseils qu'ils ont prodigués.

Je remercie également Madame Josiane MOTHE, en tant qu'encadrante de stage, pour la direction de mes recherches et son encadrement pour mener à bien ce travail de recherche de fin d'études.

J'exprime mes remerciements à Monsieur Olivier ROBINSON, mon encadrant pédagogique, pour ses conseils et sa disponibilité.

Je suis reconnaissant aux membres de jury qui ont accepté de juger mon travail.

Mes plus vifs remerciements s'adressent à ma famille pour leur soutien durant toutes mes années d'études.

Enfin, je ne saurais manquer d'exprimer ma gratitude à mes collègues et amis de la promotion de Master2 MISA 2016 pour le partage de connaissances et d'entraides.

Merci.

Table des matières

1	Introduction	2
1.1	Motivation	2
1.2	Contribution	6
1.3	Organisation du document	6
2	Etat de l'art de l'apprentissage profond	8
2.1	Généralités	8
2.1.1	Apprentissage automatique et apprentissage profond	8
2.1.2	Bref historique	9
2.1.3	Spécificité de l'apprentissage profond	10
2.2	Application dans le cadre de nos recherches	11
2.2.1	Application dans le domaine de la vision par ordinateur	11
2.2.2	Application sur le traitement automatique du langage naturel	15
2.2.2.1	Représentation vectorielle des mots	15
2.2.2.2	Modèles neuronaux pour la génération de phrases	19
2.2.3	Génération automatique de descriptions d'images	21
3	Contribution au modèle théorique	27
3.1	Présentation de la contribution	27
3.1.1	Classification des images	29
3.1.2	Génération de descriptions utilisant le vecteur de catégories	30
3.2	Implémentations	32
4	Évaluation de notre proposition	34
4.1	Ressources expérimentales	34
4.1.1	Collection de données :	34
4.1.2	Mesures d'évaluation	34
4.1.2.1	Mesures d'évaluation de la classification :	34
4.1.2.2	Mesures d'évaluation des descriptions générées :	36
4.2	Résultats	38
4.2.1	résultats de la classification	38
4.2.2	résultats de la génération de descriptions	39

5 Conclusions	41
5.1 Discussions	41
5.2 Conclusion et perspective	41

Table des figures

1.1	Illustration des différents fossés [17]	3
1.2	Apprentissage du modèle par les données d'apprentissage [17]	4
1.3	Prédiction des concepts à partir du modèle [17]	5
1.4	Exemples de description d'une image par trois légendes différentes dans Microsoft COCO Captions [Chen et al., 2015] :	5
2.1	Détection de contours par convolution discrète. Source des images : [78] . .	12
2.2	Architecture du modèle cbow de Word2Vec [46]	17
2.3	Architecture du modèle Skip-Gram de Word2Vec [46]	18
2.4	Représentation du réseau de neurones récurrents	20
2.5	Illustration des systèmes multimodaux pour l'annotation et la recherche d'images [4]	22
2.6	Illustration du modèle CNN-LSTM pour la génération de phrases décrivant l'image [8]	25
3.1	Illustration de la relation entre catégories, V_{cat} et descriptions	28
3.2	Classification multi-labels	29
3.3	Illustration du modèle de base m-RNN [42]	30
4.1	Exemples de données dans Microsoft COCO Captions [6]	35
4.2	Évolution de la performance du modèle de classification pendant l'apprentissage	38
4.3	Exemples d'images classées par le modèle	39
4.4	Evolution des mesures BLEU-1, BLEU-4, METEOR et CIDEr lors de l'apprentissage des quatre modèles	40
5.1	Architecture générale d'un réseau de neurones artificiels	44

Liste des tableaux

2.1	Tableau récapitulatif de l'architecture Alexnet [35]	13
2.2	Tableau récapitulatif de l'architecture Inception V3 [35]	14
3.1	Configuration utilisée par les quatre modèles	33
4.1	Exemples de descriptions générées	40

Acronymes

Liste des acronymes utilisés dans le rapport :

ACC (CCA) : Analyse canonique des corrélations

CBOW : Continuous Bag Of Word

CNN ou ConvNet : Convolutional Neural Network, Réseau de neurones convolutifs

gLSTM : guided LSTM

GRU : Gated Recurrent Unit

ILSVRC : ImageNet Large Scale Visual Recognition Competition

KCCA : Kernel CCA

LSTM : Long Short-Term Memory

MSCOCO : Microsoft Common Objects in COntext

m-RNN : multimodal RNN

OCR : Optical Character Recognition

RNA : Réseau de Neurones Artificiels

RNN : Recurrent Neural Network

TALN : Traitement Automatique du Langage Naturel

VO : Vision par Ordinateur

Chapitre 1

Introduction

Cette partie aborde les avantages ainsi que les problèmes rencontrés pour automatiser l'annotation d'images. Ensuite, notre méthodologie sera brièvement exposée pour aboutir à notre contribution.

1.1 Motivation

L'annotation d'images a pour but de leur attacher des informations textuelles à des images pour faciliter leurs exploitations. Par exemple, l'indexation d'une image est utilisée dans les systèmes textuels permettant à l'utilisateur de rechercher les images ; en formulant des requêtes dans un langage naturel ou pseudo naturel et exprimer ainsi ses besoins plus facilement.

L'annotation d'une image peut être effectuée soit par annotation textuelle manuelle, soit par annotation automatique basée sur le contenu de l'image.

L'annotation manuelle rencontre un problème sur le choix de termes utilisés pour annoter une image. L'annotateur a tendance à associer des termes qui lui semblent pertinents selon son interprétation mais qui sont souvent subjectifs et ambigus. L'annotation manuelle est parfois effectuée par des spécialistes, les iconographes. Ces spécialistes associent les images à des mots et groupes de mots extraits d'un thesaurus ou à des catégories prédéfinies. Ce processus est très couteux en ressource humaine et temporelle compte tenu de l'immensité et la difficulté du travail à effectuer pour une grande collection d'images, même si des applications de jeux sérieux ont été développées pour palier le problème de coût financier. En alternative, l'annotation automatique est utilisée en se basant sur le contenu de l'image.

L'annotation automatique d'images concerne, en général, l'extraction de caractéristiques visuelles de l'image jusqu'à la prédiction des concepts sémantiques les plus pertinents décrivant (par des textes) cette image.

On rencontre 3 types de problèmes, dans l'annotation automatique d'une image selon [64] [5] illustrés par la Figure 1.1. Cette figure expose en parallèle la différence entre la vision humaine et un système de vision cognitive.

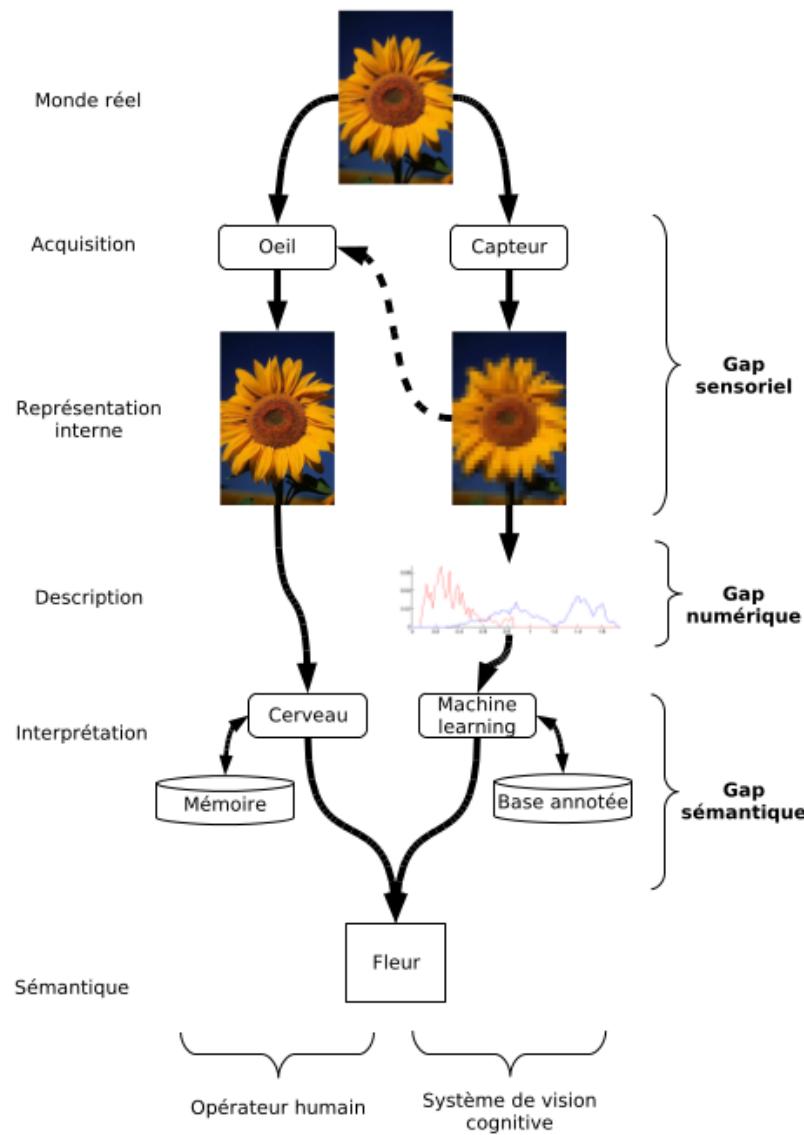


FIGURE 1.1 – Illustration des différents fossés [17]

- **Fossé sensoriel** : représente la perte et/ou la déformation des informations due aux appareils utilisés lors de l'acquisition de l'image numérique (appareil photo numérique, médical, satellitaire, scanner,...). La perte peut venir des performances de l'appareil utilisé et des bruits numériques.
- **Fossé numérique** : concerne la capacité d'un modèle (descripteur) à extraire les signatures visuelles pertinentes. Ce problème est lié au choix du descripteur. Par exemple, le choix effectué pendant le **feature engineering** (traitement des variables) se portant sur les couleurs, la forme ou la texture, les descripteurs locaux ou globaux, cela pour extraire les caractéristiques les plus pertinentes d'une image. Dans sa thèse [17], Nicolas HERVE a écrit : "Le gap numérique est l'écart entre l'information qui est présente visuellement dans une image et celle qu'un descripteur est capable d'extraire et de représenter".

- **Fossé sémantique** : problème majeur de l'annotation automatique d'images, elle peut être considérée comme le manque de corrélation entre la manière dont les humains perçoivent les informations et celle dont les ordinateurs représentent ces informations. Smeulders et al. [64] expriment ainsi : “The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation”. A la différence de l'étude des systèmes documentaires textuels, le fossé entre les caractéristiques visuelles bas niveaux de l'image (couleur, texture,...) et ses caractéristiques sémantiques de haut niveau (description et signification) est assez large.

L'annotation automatique d'une image se résume par la modélisation de la relation entre les caractéristiques visuelles de l'image et ses caractéristiques sémantiques. D'un côté, les caractéristiques visuelles d'une image numérique peuvent être extraites par des algorithmes d'analyse d'image qui étudient la distribution des valeurs de chaque pixel de l'image. D'un autre côté, l'interprétation de cette image nous conduit à la sémantique de cette image. Le défi majeur dans le cadre de l'annotation automatique concerne l'extraction automatique des informations sémantiques de l'image en réduisant la distance entre la signification et les caractéristiques visuelles qui correspondent au fossé sémantique.

Pour réduire ce fossé sémantique, des techniques d'apprentissage automatique ont été massivement utilisées et ont abouti à de bonnes performances. A partir d'une base d'apprentissage constituée d'images déjà annotées, il est possible de construire des modèles capables par la suite de prédire des annotations pour de nouvelles images [Figure 1.2 et Figure 1.3].

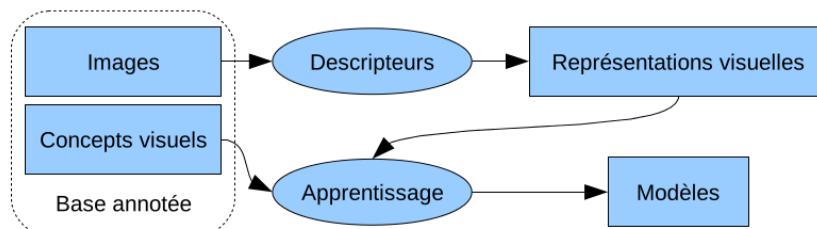


FIGURE 1.2 – Apprentissage du modèle par les données d'apprentissage [17]

Notre travail de recherche porte sur l'étude des algorithmes d'apprentissage profond pour associer images et textes à partir des réseaux de neurones artificiels (RNA). Ces modèles très récents ont été utilisés pour représenter les images et les textes afin d'extraire leurs significations et ensuite les relier.

Nous nous sommes particulièrement intéressés à la description des images par génération de phrases descriptives ou légendes (Figure 1.4). Les phrases¹ contiennent des informa-

1. Pour reprendre le vocabulaire de la littérature du domaine, nous parlerons de "phrase" correspondant à la description d'une image, même si le texte peut ne pas respecter la grammaire de la langue

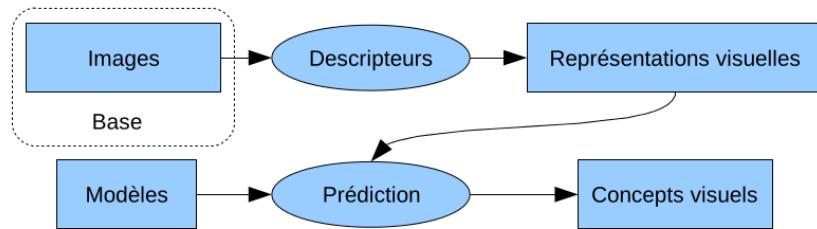


FIGURE 1.3 – Prédiction des concepts à partir du modèle [17]

tions plus détaillées des images. Elles sont composées de variétés de classes de mots : verbe, nom, adjectif,... qui suivent des règles grammaticales permettant de mieux décrire des images. Aussi dans les phrases, les concepts abstraits, comme *riding*, sont situés dans un contexte (*people* et *horse*), qui facilite leur apprentissage.



three men dressed like cowboys riding on horses.
 two people riding horses side by side through a park.
 a couple of men on horses in a field.

FIGURE 1.4 – Exemples de description d’une image par trois légendes différentes dans Microsoft COCO Captions [Chen et al., 2015] :

Une solution est d’associer directement les images et les phrases de la collection de données. Cependant, la description d’une nouvelle image nécessite des informations sur les concepts présents dans cette image pour ensuite générer de nouvelles phrases qui peuvent être différentes des phrases de cette collection. Ainsi, nous avons utilisé un modèle de génération de phrases pour la description d’une image en combinant les travaux pour la compréhension d’une image et les travaux pour la génération de phrases.

Notre proposition comprend 2 étapes : la première étape analyse les caractéristiques visuelles des images afin de les classer et la seconde fusionne les caractéristiques visuelles et textuelles pour générer des annotations.

La première étape utilise les méthodes d’apprentissage supervisé pour la classification des images. A partir des données d’apprentissage, il est possible de construire un modèle pour classifier les images dans des catégories prédéfinies. Ces catégories ou classes définissent les concepts présents dans les images. Les modèles sont créés par apprentissage

de classifieurs pour grouper les images à partir de ses caractéristiques visuelles (souvent représentés par des vecteurs) extraites des descripteurs issus des travaux dans le domaine de la vision par ordinateur comme les réseaux neuronaux convolutifs. Après avoir extrait les caractéristiques visuelles d'une nouvelle image, le modèle est utilisé pour prédire la classe correspondante. Le concept visuel associé à cette classe est ensuite attribué à cette image lors de l'annotation. Notre étude s'est focalisée sur la classification multi-classe qui a été utilisée pour la reconnaissance de formes et d'objets [35], [63], [66], [62], reconnaissance de scènes [82] dans le domaine de la vision par ordinateur.

La seconde étape de notre contribution utilise les modèles multimodaux qui exploitent la multi-modalité des données : les images et les textes associés. La tâche principale est d'analyser le contenu de l'image et de générer des descriptions textuelles en relation avec ce contenu. Les travaux effectués sont basés sur des travaux issus du domaine de la vision par ordinateur et du traitement automatique du langage naturel (TALN) pour aboutir à la génération de textes décrivant des images[51], [80],[9] [30].

1.2 Contribution

Inspirée de [79] [26], notre contribution concerne l'amélioration d'un modèle de base de génération de descriptions d'images : *m-RNN* : multimodal Recurrent Neural Network [43] [42] [41] en lui apportant des informations sémantiques additionnelles sur les images. Ces informations mettent en exergue la génération de mots relatifs à l'image à décrire. L'information sémantique est définie à partir des catégories auxquelles l'image appartient et est représentée par un vecteur de scores que nous nommons : *vecteur de catégories*.

1.3 Organisation du document

Le second chapitre aborde l'état de l'art de l'apprentissage profond et comporte deux sections :

- La section 2.1 indique les grandes lignes de l'apprentissage profond
- La section 2.2 cite les applications dans le domaine de la recherche notamment en vision par ordinateur, en traitement automatique du langage naturel et la génération automatique de descriptions d'images.

Le troisième chapitre comporte deux sections sur notre contribution pour la génération de descriptions d'images :

- La section 3.1 décrit notre contribution sur un modèle de base en utilisant les résultats issus de la classification des images.

- La section 3.2 présente les implémentations des deux parties de notre expérimentation.

Le quatrième chapitre concerne l'évaluation de notre modèle et comporte deux sections :

- La section 4.1 expose les ressources que nous avons utilisées lors de notre expérimentation à savoir la collection de données et les mesures d'évaluation du modèle.
- On retrouve dans la section 4.2 les résultats obtenus.

Le cinquième chapitre conclue le document sur une discussion globale des méthodologies et résultats obtenus et aborde les travaux en perspective.

Chapitre 2

Etat de l'art de l'apprentissage profond

Ce chapitre présente l'apprentissage profond. Les informations de base se portent sur :

- l'apprentissage profond en général dans la première section
- le suivi de ses applications dans le cadre des recherches en annotation automatique d'images dans la seconde section, à savoir : la vision par ordinateur, le traitement automatique de texte et la mise en correspondance des caractéristiques visuelles et textuelles.

2.1 Généralités

2.1.1 Apprentissage automatique et apprentissage profond

L'apprentissage profond, "*deep learning*", est une méthode d'apprentissage automatique "*machine learning*" basée sur les réseaux de neurones artificiels. L'apprentissage par expérience permet à l'ordinateur d'acquérir des connaissances dont il a besoin pour effectuer une tâche précise à partir de **données issues de phénomènes réels** sans l'intervention d'un opérateur humain.

Pour une tâche donnée, un algorithme d'apprentissage permet à un modèle d'acquérir une expérience. Cette expérience améliore sa performance pour effectuer cette tâche.

Dans le cas d'un apprentissage supervisé le modèle apprend par observation des exemples x appartenant à $\{X\}$ associés à y appartenant à $\{Y\}$. L'algorithme d'apprentissage modélise une fonction $f : X \mapsto Y$ par estimation de cette fonction. La fonction f représente la relation entre les données d'entrée et de sortie et souvent représente la loi de probabilité conditionnelle de la distribution de $y \in Y$ sachant $x \in X$: $p(y|x)$.

Pour l'apprentissage non-supervisé, le modèle est conçu pour estimer la loi de probabilité $p(x)$ de la distribution observée $\{X\}$ par : $p(x) = \prod_{i=1}^n p(x_i|x_1, \dots, x_{i-1})$

L'apprentissage profond consiste surtout en l'apprentissage de l'ordinateur pour la

compréhension d'un fait, par une représentation hiérarchique des concepts à partir de plusieurs couches. Cette représentation hiérarchique permet d'apprendre des concepts plus compliqués issus des relations entre les concepts les plus simples.

2.1.2 Bref historique

L'apprentissage profond a débuté dans les années 40. Il est l'aboutissement de l'évolution du domaine de Réseau de Neurones Artificiels (RNA)[22].

- **Cybernetics (1940 à 1960)** : a été le premier précurseur des modèles linéaires. Cette époque a été marquée par un modèle appelé "*perceptron*" fabriqué par Rosenbatt en 1958 [57] qui a été inspiré par le travail de McCulloch et Pitts dans [44] sur l'étude biologique de l'apprentissage. Grâce aux perceptrons ils ont pu implémenter un modèle capable de classifier une entrée dans 2 catégories (1 ou 0).
- **Connexionnisme (1980 à 1990)** : Depuis a émergé la science cognitive qui est un domaine pour l'étude de la pensée. Le connexionnisme est basé principalement sur le fait que l'interaction entre plusieurs neurones accroît l'intelligence. Une des grandes découvertes dans ce domaine est la rétropropagation [16] qui est largement utilisée pour l'apprentissage des RNA.
- **Deep learning ou apprentissage profond (2006 à aujourd'hui)** : s'inspire de plusieurs domaines, pour améliorer les systèmes existants et créer des modèles profonds, spécialement les mathématiques appliquées : l'algèbre linéaire, la probabilité, la théorie de l'information et l'analyse numérique.

En 1998, Yan LeCun et al. ont déployé le premier système utilisant un modèle d'apprentissage profond LeNet-5 [36]. LeNet-5 a été modélisé pour la reconnaissance de l'écriture manuscrite (OCR) et intégré dans un système pour la reconnaissance de documents. Dans [36] Yan LeCun et al. décrivent LeNet-5 où ils évoquent une technique d'apprentissage nommé : GTN (Graph Transformer Networks). Ces derniers ont utilisé un réseau de neurones convolutif ou CNN (Convolutional Neural Network ou ConvNet) entraîné par un algorithme de descente de gradient : la rétropropagation. Ce modèle est aujourd'hui la base de la majorité des modèles d'apprentissage profond surtout appliqués à la vision par ordinateur.

Grâce aux larges données d'apprentissage, les chercheurs ont pu entraîner des modèles de réseau de neurones profond¹ en évitant le sur-apprentissage.

En 2012, Alex Krizhevsky et al.[35] ont remporté la première place lors d'un concours organisé par ImageNet : ImageNet ILSVRC en 2012 grâce à leur réseau de neurones convolutif profond : AlexNet. En résumé, AlexNet est composé de cinq couches de convolution et trois couches interconnectées utilisées comme classifieur. Pour ce modèle, on compte un

1. la profondeur d'un réseau est définie par le nombre de couches formant le réseau d'où le nombre de paramètres

total de 650.000 neurones et 60 millions de paramètres et il a servi à classifier 1,2 millions d'images suivants 1000 classes différentes issues de l'ImageNet ILSRVC-2010. Ce modèle a été entraîné dans un temps raisonnable (5 à 6 jours) en assignant le traitement de calculs à des cartes graphiques programmables.

Les grandes collections de données et les cartes graphiques programmables² ont rendu l'apprentissage profond plus accessible et ont contribué à l'accroissement des travaux dans ce domaine.

2.1.3 Spécificité de l'apprentissage profond

L'apprentissage automatique s'intéresse surtout aux problèmes et tâches qui sont subjectifs et qui ne peuvent pas être décrits précisément par un langage formel. Ces tâches sont trop complexes pour y appliquer des règles logiques et les programmer "en dur" [*hard coding*]. Selon wikipédia [77] : "La difficulté réside dans le fait que l'ensemble de tous les comportements possibles compte tenu de toutes les entrées possibles devient rapidement trop complexe à décrire (on parle d'explosion combinatoire) dans les langages de programmation disponibles. On confie donc à des programmes le soin d'ajuster un modèle permettant de simplifier cette complexité et de l'utiliser de manière opérationnelle." Comme exemple, on peut citer les problèmes de perception : reconnaissance d'objets et de formes ou vocales.

Un des avantages d'utiliser l'apprentissage profond par rapport aux autres algorithmes d'apprentissage est l'extraction des caractéristiques pertinentes des données pour résoudre le problème étudié. Cette étape est généralement effectuée lors d'un "*feature engineering*" dans l'apprentissage automatique classique. En apprentissage profond, cette étape est assignée à l'algorithme lui-même par "***feature learning***" (apprentissage de la représentation). Un autre problème rencontré lors d'un apprentissage automatique est l'identification des sources qui influencent les valeurs des données observées : **facteurs de variation**. Ces facteurs ne sont pas quantifiables et sont souvent difficiles à identifier. L'apprentissage profond introduit alors la représentation hiérarchique de l'entrée sur plusieurs couches du réseau. Les caractéristiques extraites des neurones d'une couche précédente sont alors pondérées et partagées pour former des caractéristiques plus complexes. Ainsi on peut facilement établir une représentation plus complexe à partir de la combinaison des caractéristiques issues des couches antérieures en augmentant la profondeur du réseau c'est-à-dire en ajoutant des couches supérieures.

Le principal enjeu est d'agrégner les variables et ses interactions dans les systèmes complexes en créant un modèle de basse dimension qui permet d'expliquer ces systèmes. L'apprentissage profond permet de généraliser les problèmes. Grâce à l'apprentissage hiérarchique des invariants qui sont caractéristiques du problème, ces modèles évoluent facilement en fonction des tâches auxquelles ils sont soumis.

2. GPUs : qui à la fois permettent les calculs en parallèle et atteignent les trillions de calculs par seconde

2.2 Application dans le cadre de nos recherches

Un algorithme d'apprentissage profond a pour objectif de créer un modèle représentant une fonction ou système complexe pour avoir une meilleure représentation des données les plus complexes grâce à un réseau de neurones artificiels (RNA).

Les réseaux de neurones artificiels ont la capacité de modéliser une fonction f linéaire ou non-linéaire. La combinaison de couches linéaires qui effectuent des translations sur leurs entrées et des couches non-linéaires définies par la fonction d'activation non-linéaire de chaque neurone (ex : ReLu, Tanh) permet d'approximer toute fonction³.

En général, un réseau de neurones profond est défini par une succession de plusieurs couches faisant intervenir plus de paramètres qui nous autorise à approximer les fonctions les plus complexes. Les informations générales concernant les réseaux de neurones artificiels et leur apprentissage sont détaillées dans l'**Annexe**.

Les sections suivantes présentent les travaux étudiés dans le domaine de l'apprentissage profond. Ces travaux concernent l'application de l'apprentissage profond dans le domaine de la vision par ordinateur (2.2.1), le traitement automatique du langage naturel (2.2.2) et l'appariement des images et textes (2.2.3).

2.2.1 Application dans le domaine de la vision par ordinateur

L'utilisation de l'apprentissage profond dans le domaine de la vision par ordinateur est la reconnaissance d'objets et de formes à partir des réseaux de neurones convolutifs ou Convolutional Neural Network (CNN) en anglais.

La reconnaissance a pour objectif de développer un algorithme qui soit capable de recevoir en entrée une image et de faire sortir la classe des objets (dans l'image) si on parle de catégorie d'objets et la référence d'un objet précis si on parle d'instance d'objets. La difficulté de la reconnaissance visuelle par un ordinateur réside sur la variation extrême de la forme et de l'apparence des objets d'une classe (par exemple un ordinateur) dans le monde réel. En effet, si l'apparence n'était pas aussi variable, il suffirait d'effectuer une correspondance exhaustive à une base de données, ce qui n'est pas le cas.

Un réseau de neurones convolutifs est une succession de couches de neurones appliquant à chaque entrée l'opérateur convolution pour exploiter la structure spatiale des données en entrée (par exemple une matrice de pixels pour les images). La convolution permet d'extraire les caractéristiques d'une image numérique en appliquant le filtre (*convolution kernel*) suivant les axes possibles : hauteur, largeur et la profondeur (les canaux rouge, bleu et vert) de cette image. Une couche de convolution utilise la convolution discrète pour une transformation linéaire des valeurs des pixels en préservant l'ordre de ces valeurs⁴

3. Borel mesurable, toute fonction de projection d'une espace discrète de dimension finie à une autre. (voir la théorie de l'approximation universel cité dans [2])

4. chaque pixel étant fortement corrélé avec les pixels voisins

dans l'espace local de l'image . Le filtre varie en fonction des effets désirés. La figure 2.1 illustre une détection de contours sur l'image en entrée par convolution. Les réseaux de neurones convolutifs adoptent des principes importants pour la représentation des entrées : la connectivité locale et le partage des paramètres.

La connexion locale des neurones des couches adjacentes assure que les filtres appris produisent des caractéristiques locales les plus fortes à un motif localisé. Précisément, chaque neurone n'est connecté qu'à une sous-région (champ réceptif local) correspondant à un certain nombre de neurones voisins dans la couche précédente.

Le partage des paramètres utilisés par l'opération de convolution signifie que, plutôt que d'apprendre un ensemble distinct de paramètres pour chaque emplacement, le modèle apprend un seul ensemble. Ainsi tous les neurones dans une couche de convolution donnée détectent exactement la même caractéristique.

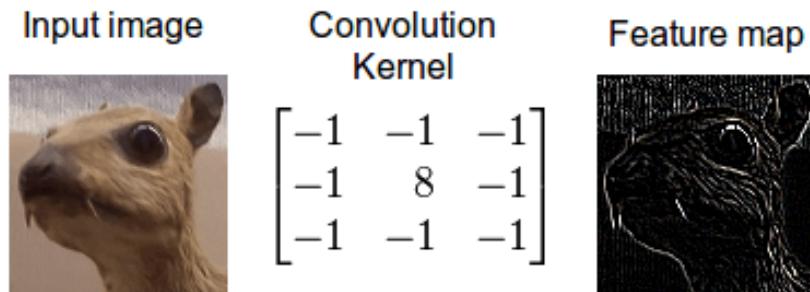


FIGURE 2.1 – Détection de contours par convolution discrète. Source des images : [78]

Ce type de modèle est utilisé pour la classification d'images car il fournit une bonne représentation de l'image par représentation hiérarchique issue des différentes couches de convolution et des couches de mise en commun (**pooling**) consécutifs. Cette architecture permet de réduire la sensibilité en translation, rotation et échelle ou aux faibles déformations de la représentation apprise.

Dans les paragraphes suivants nous décrivons la structure des modèles utilisés pour la classification d'images que nous avions étudiées dans nos travaux.

AlexNet [35]

Ce modèle a été établi par Alex Krizheski et al. Le modèle est composé de 9 couches successives (voir tableau 2.1) dont cinq sont des couches de convolution pour l'extraction des caractéristiques de l'image et trois sont des couches entièrement connectées pour la classification de cette image.

AlexNet utilise deux techniques pour l'accélération de l'apprentissage par rétropropagation et éviter le sur-apprentissage : ReLu ou Rectified Linear Unit comme fonction d'activation des neurones et Local Response Normalization pour la normalisation.

TABLE 2.1 – Tableau récapitulatif de l'architecture Alexnet [35]

Couche	Entrée	Filtres (nombre – dimensions – pas) ou Nombres de sortie
Première	Image de $224 \times 224 \times 3$	$96-11 \times 11 \times 3-4$
Seconde	Sortie 1-ère couche + Normalisation et pooling	$256 - 5 \times 5 \times 48$
Troisième	Sortie 2-nde couche + Normalisation et pooling	$384 - 3 \times 3 \times 256$
Quatrième	Sortie 3-ème couche	$384 - 3 \times 3 \times 192$
Cinquième	Sortie 4-ème couche	$256 - 3 \times 3 \times 192$
3 couches interconnectées : FC6 - FC7 - FC8	Sortie 5-ème couche + Normalization et pooling	4096
Une couche de sortie pour les 1000 classes	Sortie fully connected layer + Softmax	1000

TABLE 2.2 – Tableau récapitulatif de l'architecture Inception V3 [35]

Type	Taille entrées
Convolution	$299 \times 299 \times 3$
Convolution	$149 \times 149 \times 32$
Convolution padded	$147 \times 147 \times 32$
Pooling	$147 \times 147 \times 64$
Convolution	$73 \times 73 \times 64$
Convolution	$71 \times 71 \times 80$
Convolution	$35 \times 35 \times 192$
$3 \times$ inception	$35 \times 35 \times 288$
$5 \times$ inception	$17 \times 17 \times 768$
$2 \times$ inception	$8 \times 8 \times 1280$
Pooling	$8 \times 8 \times 2048$
Linear	$1 \times 1 \times 2048$
softmax	$1 \times 1 \times 1000$

VGG - OxfordNet [63]

VGG ou OxfordNet sont des modèles créés par un groupe de l'université d'Oxford : Visual Geometry Group. Le groupe ont déployé ses deux meilleurs modèles : VGG-16 (16 couches) et VGG-19 (19 couches).

GoogLeNet [66] et Inception

Christian Szegedy et al. dans [66] ont présenté une autre architecture nommée GoogleNet. Leur contribution a été de créer un modèle plus profond (22 couches) pour améliorer les performances des réseaux profonds tout en réduisant le coût du traitement. Le module "inception" a été alors créé qui est une combinaison de convolutions (de filtre 1×1 , 3×3 et 5×5) et de pooling en parallèle.

L'inception a été amélioré par Christian et Serguei présenté dans [23] et [67]. La première amélioration [23] a été la normalisation des données pour chaque lot : batch-normalized inception ou inception V2. La normalisation consiste à centrer et réduire les entrées pour chaque couche. L'architecture de la seconde amélioration inception V3 [67] est présentée dans le tableau 2.2.

Ces modèles ont été entraînés pour classifier des millions d'images de la collection ILSVRC ou ImageNet Large Scale Visual Recognition Challenge [58] sur 1000 classes différentes.

Ces modèles pré-entraînés peuvent être utilisés pour effectuer des tâches suffisamment similaires (par exemple classification d'autres classes que les classes d'ILSRVC [39]) et accélèrent l'apprentissage du réseau par transfert des paramètres. Pour adapter ces modèles aux nouvelles tâches on procède alors à une *fine-tuning* ou *réglage fin* du modèle : classification de scènes [68] [82], détection d'objets [28], [56],[74]. Ces modèles sont efficaces du fait qu'ils ont été entraînés sur un large ensemble de données de plusieurs classes pour détecter les caractéristiques et structures pertinentes des images.

Ces modèles sont aussi intéressants car, en tant que **descripteur**, ils permettent l'extraction des vecteurs caractéristiques des images ou vecteurs descripteurs. Des vecteurs caractéristiques peuvent être extraits à partir de ces modèles pour représenter les images en entrée et les utiliser dans d'autres applications plus complexes.

2.2.2 Application sur le traitement automatique du langage naturel

Les problèmes étudiés dans le traitement automatique du langage naturel se concentrent sur une meilleure représentation des textes (mots, groupes de mots, phrases) pour apporter une connaissance à l'ordinateur sur leur signification et leurs propriétés en vue d'effectuer des tâches du TALN comme la recherche de mots clés, recherche de synonymes, la traduction,... . A la différence de certaines méthodes considérant chaque texte comme une combinaison possible des lettres de l'alphabet, les travaux suivants concernent une représentation par des vecteurs. La représentation vectorielle facilite la résolution des problèmes d'analyse en utilisant les différentes propriétés et opérations vectorielles, comme le calcul de la distance vectorielle.

2.2.2.1 Représentation vectorielle des mots

Une première représentation est la représentation par « sac-de-mots » ou bag of word en anglais. La représentation est basée sur la matrice de cooccurrence, qui est constituée par la fréquence des mots dans chaque document. La matrice est donc obtenue en comptant le nombre de fois où le mot apparaît dans chaque document. Chaque mot est alors représenté par la ligne correspondante de la matrice de cooccurrence et le document par la colonne correspondante. Cette représentation en grande dimension ne capture que peu d'informations sur la signification de chaque mot.

D'autre part, la signification d'un mot est en référence avec les mots qui l'entourent. Les méthodes sont fondées par la constatation suivante : « Les mots qui apparaissent dans un même contexte ont tendance à avoir les mêmes significations » comme le formulait

Harris [15]. Ainsi Sahlgren M., [59] a proposé « **l'hypothèse de distribution** » pour quantifier la signification des mots en vue d'extraire leurs relations.

Les approches que nous avons étudiées concernent les méthodes de *"word embedding"* utilisant les réseaux de neurones artificiels (RNA).

Les *word embedding* sont des méthodes qui permettent de représenter les mots par des vecteurs à dimensions réduites (200-1000 mais largement inférieures à la taille du vocabulaire) en modélisant la relation entre les mots : sémantique et syntaxique. On définit le *word embedding* par la projection des mots dans un espace vectoriel :

$$\text{word-embedding} : W : \text{mots} \mapsto \mathbb{R}^n$$

Pour les mots similaires (apparaissant dans le même contexte) w_s $W(w_s)$ sont plus proches (suivant la mesure de distance utilisée dans l'espace).

Cette représentation continue permet aussi de capturer des analogies entre les mots. Par exemple, les notions de genre et nombre en utilisant les opérateurs arithmétiques vectoriels :

$$\begin{aligned} W('woman') - W('man') &\simeq W('aunt') - W('uncle') \\ &\simeq W('queen') - W('king') \end{aligned}$$

Pour cette représentation distributive des mots à partir des RNA, Mikolov et al. [46] ont proposés deux 2 modèles n-gramme appelés *"modèles word2vec"* : skip-gram et cbow (continuous bag of words). Skip-gram et cbow sont 2 architectures de RNA simples.

En général, les modèles word2vec sont entraînés pour prédire les mots voisins et obtenir une meilleure représentation des mots. Ces architectures construisent des fenêtres de contexte obtenues en prenant les C mots qui précédent et les C mots qui suivent un mot appelé **mot central**.

cbow : L'architecture du modèle cbow est illustré par la figure 2.2. Pour calculer le vecteur représentatif des mots de la vocabulaire, le modèle prend en entrée une représentation en sac de mot binaire du mot du contexte $x = x_1, \dots, x_v$. W est la matrice des paramètres, de dimension $V \times N$, entre la couche d'entrée et la couche cachée. Chaque ligne de la matrice W est un vecteur représentatif v_{wI} du mot en entrée associé wI . On obtient la sortie de la couche cachée, h , par :

$$h = W^T x := v_{wI}^T \quad (2.1)$$

On défini par la matrice W' ($N \times V$) les paramètres entre la couche cachée et la couche de sortie. W' est utilisé pour calculer le score u_j pour tous les mots du vocabulaire, donné par :

$$u_j = v_{wj}^{T'} h \quad (2.2)$$

v_{wj}' est le j-ème colonne de la matrice W' .

Le score est ensuite normalisé par la fonction *softmax* pour obtenir la probabilité post-

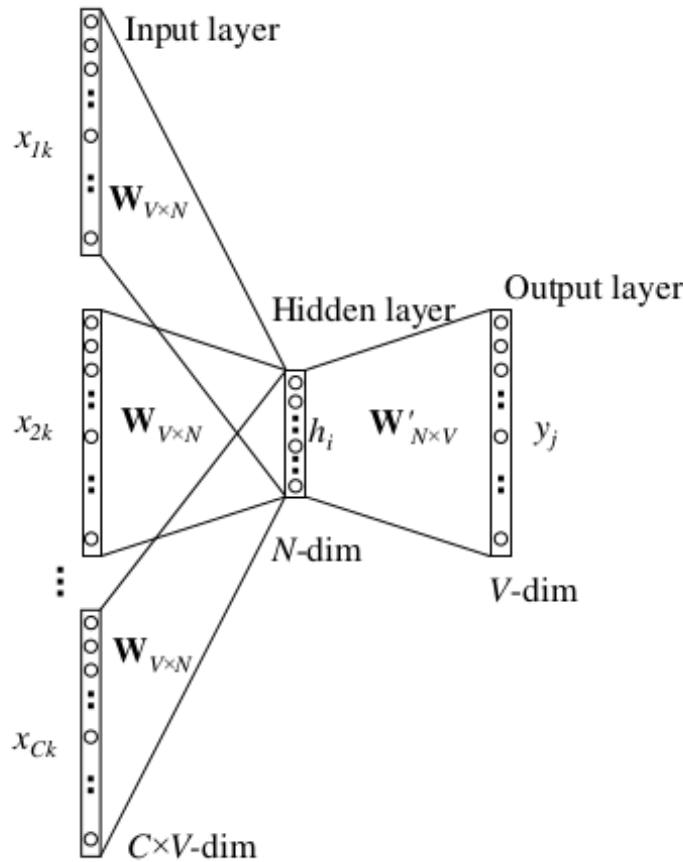


FIGURE 2.2 – Architecture du modèle cbow de Word2Vec [46]

priori :

$$p(w_j|w_I) = y_j = \frac{\exp(u_j)}{\sum_{j'=1}^V \exp(u_{j'})} \quad (2.3)$$

v_w et v'_w sont deux représentations du mot w .

Dans sa version finale, le modèle cbow prend en entrée une fenêtre de mots de contexte W_1, \dots, w_C . C est le nombre de mots dans le contexte. Ainsi, on redéfinit la sortie de la couche cachée par l'équation :

$$h = \frac{1}{C} W^T (x_1 + x_2 + \dots + x_C) \quad (2.4)$$

$$= \frac{1}{C} (v_{w1} + v_{w2} + \dots + v_{wC} C^T) \quad (2.5)$$

v_w est le vecteur associé au mot w .

skip-gram : Skip-gram (figure 2.3) est le modèle opposé à cbow. Les mots dans la fenêtre de contexte se placent sur la couche de sortie pour être prédits et le mot central sur la couche en entrée. Le mot central, représenté en sac-de-mo binaire, et projeté dans la couche cachée par la matrice des paramètres W .

$$h = W_{(h,.)}^T := v_{wI}^T \quad (2.6)$$

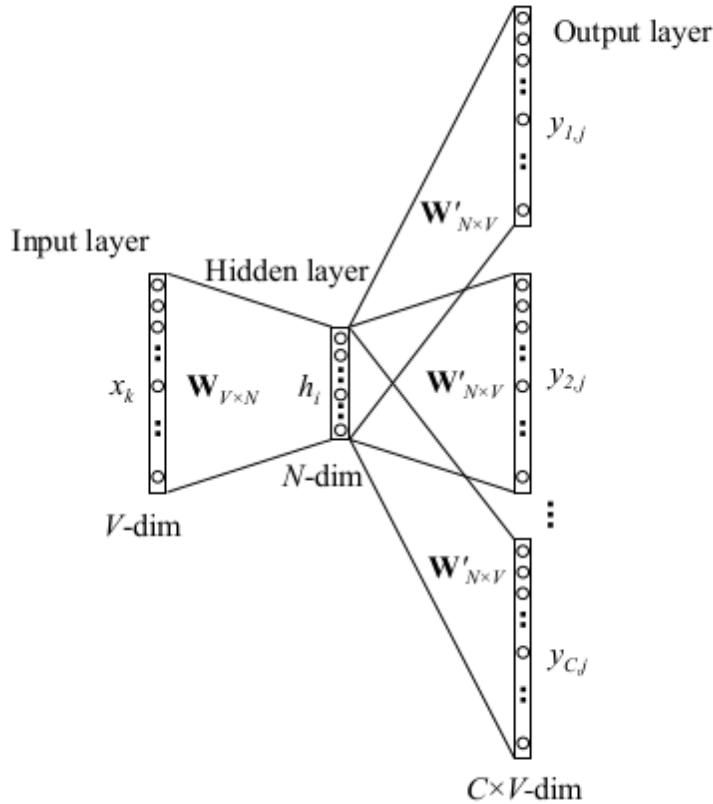


FIGURE 2.3 – Architecture du modèle Skip-Gram de Word2Vec [46]

Pour C mots dans la fenêtre de contexte, on estime la distribution multinomiale à partir de la fonction *softmax* par :

$$p(w_{c,j}|w_I) = y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{j'=1}^C \exp(u_{j'})} \quad (2.7)$$

$w_{c,j}$ est j-ème mot de la c-ème fenêtre de la couche de sortie.

w_I est le mot central en entrée.

$y_{c,j}$ est la sortie de la j-ème neurone et c-ème fenêtre de la couche de sortie par *softmax*.

$u_{c,j}$ est l'entrée de la j-ème neurone et c-ème fenêtre de la couche de sortie.

La méthode par descente du gradient est utilisée pour entraîner les modèles à trouver les matrices de projection : word embedding et contexte embedding qui maximisent la probabilité de la similarité entre le mot central et les mots du contexte. Concrètement, le modèle est entraîné en maximisant les fonctions log-vraisemblance (dans tout le vocabulaire) données par les formules suivantes :

$$\frac{1}{T} \sum_{t=1}^T \log p(w_t|w_{t-\frac{c}{2}} \dots w_{t+\frac{c}{2}}) \quad (2.8)$$

pour le modèle cbow

$$\frac{1}{T} \sum_{t=1}^T \sum_{j=t-c, j \neq t}^{t+c} \log p(w_j | w_t) \quad (2.9)$$

pour le modèle skip-gram

T : taille des données d'apprentissage

c : taille maximum de la fenêtre de contexte

2.2.2.2 Modèles neuronaux pour la génération de phrases

Dans cette partie une phrase est traitée en tant que séquence de mots ordonnés. La génération d'une phrase est plus complexe car elle est déterminée par les représentations des expressions (mots) qui les constituent et les règles de grammaire utilisées pour les combinés.

Les modèles de langue neuronaux sont utilisés pour prédire une séquence de mots dans les systèmes de reconnaissance vocale, la traduction automatique et les systèmes questions-réponses. Ces modèles sont entraînés pour estimer la loi de probabilité de la séquence de m mots $p(w_1, \dots, w_m)$ par apprentissage des réseaux de neurones.

Dans les modèles n-grams, cette probabilité est conditionnée par une fenêtre des n mots (n fixe).

$$p(w_1, \dots, w_m) = \prod_{t=1}^m p(w_t | w_1, \dots, w_{t-1}) \approx \prod_{t=1}^m p(w_t | w_{t-(n-1)}, \dots, w_{t-1}) \quad (2.10)$$

$w_{t-(n-1)}, \dots, w_{t-1}$: représente le contexte.

Ces modèles sont limités à la représentation de séquences car ils sont conditionnés par la fréquence des cooccurrences des n-grammes dans le document. Il est rare que des n-grammes (pour n assez grand) correspondent exactement dans des phrases similaires vu *l'infinie* des variations possibles. Ainsi, les réseaux de neurones récurrents ont été introduits [48] [49]. L'avantage d'utiliser les réseaux de neurones récurrents, par rapport aux réseaux de neurones simples, est leur capacité à capturer une meilleure représentation des séquences de données de tailles variables [60].

Grâce à la représentation de l'historique des mots à partir de la couche cachée, les réseaux de neurones récurrents permettent de modéliser cette probabilité en prenant en compte tous les mots précédents (de nombre variable) sous forme de contexte.

A un instant t , les paramètres des réseaux de neurones récurrents sont définis par :

$$h_t = \sigma(W^{(hh)} h_{t-1} + W^{(hx)} x_t) \quad (2.11)$$

$$\hat{y}_t = \text{softmax}(W^{(S)} h_t) \quad (2.12)$$

$x_t \in \mathbb{R}^d$ est le vecteur (sac-de-mot binaire) qui représente le mot courant à l'instant t

h_t : est la sortie de la couche cachée

$W^{(hh)} \in \mathbb{R}^{D_h \times D_h}$ sont les paramètres qui conditionnent la sortie de la couche cachée à l'instant précédent $t - 1$

$W^{(hx)} \in \mathbb{R}^{D_h \times d}$ est la matrice de projection de l'entrée. Elle correspond à la matrice de *word embedding* citée en ??.

σ est une fonction non-linéaire (par exemple sigmoid)

$\hat{y}_t \in \mathbb{R}^{|V|}$ permet de générer le mot suivant de la séquence observée sachant le contexte (à partir de h_{t-1}) et le mot observé (représenté par x_t), ceci en générant la probabilité pour chaque mot du vocabulaire à l'instant t : $\hat{p}(x_{t+1} = v_j | x_t, \dots, x_1) = \hat{y}_{tj}$

$W^{(S)} \in \mathbb{R}^{|V| \times D_h}$

$|V|$ est la taille du vocabulaire

Le réseau est entraîné par la maximisation de vraisemblance en minimisant la fonction de l'entropie croisée à l'instant t (sommée sur tout le vocabulaire)[equation 2.13] par rétropropagation à travers le temps [76].

$$J^{(t)}(\theta) = - \sum_{j=1}^{|V|} y_{tj} \times \log(\hat{y}_{tj}) \quad (2.13)$$

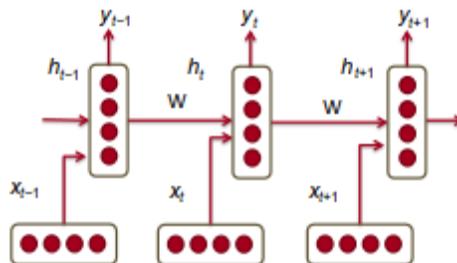


FIGURE 2.4 – Représentation du réseau de neurones récurrents

Le but d'un réseau neuronal récurrent est de propager le contexte étape par étape, cependant cette technique présente des défauts sur des séquences assez longues. Le problème se rapporte à la disparition (ou explosion) du gradient lors de la rétropropagation qui compromet la performance de ces modèles [18].

De nouveaux types de réseaux de neurones récurrents plus performants ont émergé pour contourner le problème de la disparition du gradient à savoir : les GRU : *Gated Recurrent Units* et LSTM : *Long-Short-Term-Memories* [19]. Ces derniers sont des extensions des réseaux de neurones récurrents simples en utilisant des unités d'activation plus complexes. Ils sont conçus de manière à avoir une mémoire plus persistante pour faciliter la capture des dépendances à long terme. Dans tout ce qui suit, la représentation de ces réseaux récurrents est tirée de <http://cs224d.stanford.edu> qui est plus élaborée et plus compréhensible.

Le modèle GRU est composé des unités (*gates*) suivants :

$$\begin{aligned}
 z_t &= \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) && \text{update gate} \\
 r_t &= \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) && \text{reset gate} \\
 \check{h}_t &= \tanh(r_t \circ Uh_{t-1} + Wx_t) && \text{Newmemory} \\
 h_t &= (1 - z_t) \circ \check{h}_t + z_t \circ h_{t-1} && \text{hidenstate}
 \end{aligned}$$

Mathématiquement, les unités d'un LSTM est définie par :

$$\begin{aligned}
 i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) && \text{input gate} \\
 f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) && \text{forget gate} \\
 o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) && \text{output gate} \\
 \check{c}_t &= \tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) && \text{new memory cell} \\
 c_t &= f_t \circ c_{t-1} + i_t \circ \check{c}_t && \text{final memory cell} \\
 h_t &= o_t \circ \tanh(c_t) && \text{hidenstate}
 \end{aligned}$$

Avec ses unités, un LSTM est capable d'apprendre des séquences d'informations assez longues et complexes.

Ces modèles de langue neuronaux sont utilisés dans le TALN : les systèmes de traduction, la reconnaissance automatique de la parole et à la description des images [41] [48], [49],[47] [13] [65]

2.2.3 Génération automatique de descriptions d'images

Cette partie du rapport s'intéresse aux modèles multimodaux pour l'appariement de texte et images et à la génération automatique de descriptions.

Présenté dans les sections précédentes, l'apprentissage profond est un outil performant dans le domaine de la vision par ordinateur et le TALN. Les modèles de génération automatique de descriptions étudiés combinent ces méthodes pour analyser d'un côté les images et de l'autre les textes afin de les associer.

En général, les modèles analysent les propriétés statistiques de chaque modalité (texte et image) de la base de données d'apprentissage. Ces méthodes ont pour objectif de projeter les caractéristiques visuelles et textuelles dans un même espace (espace d'intégration ou espace sémantique). Ces systèmes sont utilisés à la fois pour la rechercher des textes associés à une image et vice-versa (figure 2.5). Pendant l'apprentissage, les images et textes sont projetés dans l'espace d'intégration de telle sorte que les plus proches voisins ont des significations assez similaires.

L'ACC (Analyse Canonique de corrélation) a été utilisée dans nombreuses études pour explorer les relations pouvant exister entre deux variables aléatoires de dimension différente (littérature ACC on parle de vue). Dans notre étude les deux variables

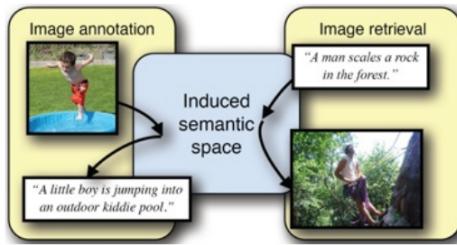


FIGURE 2.5 – Illustration des systèmes multimodaux pour l’annotation et la recherche d’images [4]

correspondent aux vecteurs caractéristiques des deux modalités. Le but de l’ACC est de trouver 2 vecteurs de projections w_x et w_y tels que la corrélation entre la projection des 2 variables $X \in \mathbb{R}^{m \times p}$ et $Y \in \mathbb{R}^{q \times m}$ (d’un échantillon de taille m) soit maximisée. Ces 2 vecteurs de projections w_x et w_y sont calculés par maximisation du coefficient de corrélation ρ qui se réduit par :

$$\rho = \arg \max_{w_x, w_y} \frac{w_x^T X Y^T w_y}{\sqrt{(w_x^T X X^T w_x)(w_y^T Y Y^T w_y)}} \quad (2.14)$$

Les vecteurs de projection maximisant le coefficient de corrélation ρ sont alors utilisés pour projeter les vecteurs des 2 vues afin de les comparer.

Cette méthode a été utilisée par [51] pour l’annotation des images par des labels. Dans leur article, des variations de l’ACC ont été aussi étudiée à savoir Kernel CCA. Le Kernel CCA se différencie de ACC par la projection des variables dans un espace de grande dimension appelé : espace de redescription [kernel trick], une transformation non-linéaire pour exploiter des relations non-linéaires entre les variables. Les vecteurs caractéristiques X de l’image et Y des labels ou étiquettes ont été respectivement extraits grâce à un réseau de neurones convolutif pré-entraîné d’Oxford (VGG) et le modèle skip-gram pré-entraîné de Mikolov et al. Word2Vec [46]. [12] [11] a montré qu’une normalisation appropriée améliore l’ACC linéaire sur une grande collection de données. L’apprentissage profond intervient dans l’ACC en proposant le Deep CCA [45]. L’efficacité de l’apprentissage profond à maximiser la corrélation définie par l’ACC a été démontrée et ceci par apprentissage bout à bout à travers un réseau de neurones par rétropropagation.

A part l’utilisation de l’ACC les modèles profonds traitent le problème par apprentissage profond de la similarité. Les paramètres de projection de différentes modalités sont calculés par apprentissage d’un RNA en optimisant une fonction objectif convenable à l’appariement des modalités dans l’espace sémantique. Ces modèles sont entraînés par rétropropagation des erreurs et s’adaptent facilement à de grandes quantités de données.

DeViSE : Deep Visual-Semantic Emedding est un modèle créé par Andrea Frome et al. [10] pour la classification d'images sur beaucoup de catégories. *DeViSE* utilise les textes associés à l'image pour améliorer les systèmes de classification existants. La contribution majeure de ce modèle est l'apprentissage des réseaux de neurones convolutifs à prédire les vecteurs caractéristiques textuels des labels associés à l'image en entrée. Pour cela Andrea Frome et al. ont pré-entraîné un réseau de neurones convolutifs pour la reconnaissance d'objets basé sur l'architecture d'AlexNet. Les labels associés à chaque image sont représentés par le vecteur représentatif issu des méthodes de word embedding à partir du modèle de Mikolov et al. entraîné sur 5.7 millions de documents (5.4 billion mots) issus du wikipedia.org. Le réseau de neurones convolutifs pré-entraîné a été ensuite modifié pour prédire les vecteurs caractéristiques de chaque texte associé aux images par apprentissage en optimisant la fonction objectif à marge suivante :

$$cout(image, label) = \sum_{j \neq label} \max[0, marge - \vec{t}_{label} M \vec{v}(image) + \vec{t}_j M \vec{v}(image)] \quad (2.15)$$

$\vec{v}(image)$ est le vecteur caractéristique de l'image en entrée

M est la matrice des paramètres.

\vec{t}_{label} est le vecteur caractéristique du label associé à l'image

Les \vec{t}_j sont les vecteurs caractéristiques des autres labels.

Cette approche a permis à ce modèle de classifier des images appartenant à d'autres catégories auxquelles le modèle n'a pas été entraîné. Cela est dû, en majeure partie, à l'utilisation des vecteurs caractéristiques issus du skip-gram modèle de Mirkov et al.

Plusieurs travaux s'inspirent de DeViSE pour créer des modèles pour associer image et texte.

Certains travaux ont proposé des fonctions objectifs plus intéressantes : les fonctions objectifs bidirectionnelles (bidirectional ranking loss). En plus d'encourager l'attribution de scores supérieurs aux phrases décrivant l'image, ces fonctions assurent pour chaque phrase que : les images qu'elle décrit aient des scores supérieurs à ceux des images décrites par les autres.

Dans [30] et [32] Andrej Karpathy et Li Fei-Fei ont défini une fonction objectif structurée pour aligner des fragments d'image (régions de l'image) et des fragments de texte (groupe de mots) pour générer des descriptions pour chaque région pertinente de l'image. Cette fonction objectif entraîne le modèle pour que les scores obtenus par les images et textes correspondants aient des scores largement supérieurs (à l'aide d'une marge) à ceux qui ne se correspondent pas (bidirectionnelle). Le score obtenu à partir d'une image k et d'une phrase l est donné par l'équation 2.16.

$$S_{kl} = \sum_{t \in g_l} \sum_{i \in g_k} \max(0, v_i^T s_t) \quad (2.16)$$

g_k est l'ensemble des fragments de l'image k et g_l l'ensemble des fragments du texte l .

$v_i^T s_t$ est le produit scalaire, interprété comme étant la mesure de similarité entre le i-ème

région de l'image et le t-ème mot de la phrase.

v_i : projection du vecteur caractéristique de l'i-ème région de l'image.

Les régions ont été extraites à partir d'un modèle de réseau neuronal convolutif utilisé dans la détection d'objets [28] pré-entraîné sur ImageNet et affiné pour la détection de 200 classes d'ImageNet Detection Challenge [58].

s_t : est la projection du vecteur caractéristique du t-ème mot dans le modèle de langue [30] [32].

Ainsi tous les mots s_t sont alignés par la meilleure région de l'image. En considérant que $k = l$ désigne la correspondance entre l'image et la phrase, la fonction objectif structurée finale est définie par :

$$C(\theta) = \sum_k [\sum_l \max(0, S_{kl} - S_{kk} + 1) + \sum_l \max(0, S_{lk} - S_{kk} + 1)] \quad (2.17)$$

De même, pour l'apprentissage de leur modèle, Ryan Kiros et al. [33] ont minimisé la fonction objectif bidirectionnelle suivante pour apparter les phrases et images qui se correspondent.

$$C(\theta) = \sum_x [\sum_k \max(0, \alpha - s(x, v) + s(x, v_k) + \sum_v \sum_k \max(0, \alpha - s(v, x) + s(v, x_k))] \quad (2.18)$$

$s(a, b) = a.b$ est le produit scalaire entre le vecteur a et b

v_k sont les vecteurs caractéristiques des phrases qui ne correspondent pas (ne décrivent pas) l'image x .

x_k sont les images qui ne sont pas décrites par v .

α : est la marge (=1 pour les méthodes citées dans [30] et [32])

Récemment, les travaux de recherche s'intéressent à la génération de descriptions d'images par des phrases : "*image captionning*" en anglais. Les modèles pour la génération de phrases descriptives suivants appliquent le principe de l'autoencodeur : encodeur-décodeur, issus des systèmes de traduction neuronaux (Neural Machine Translation), sur les images et les phrases associées. Dans ce cas, l'encodeur concerne l'analyse et la représentation des images et le décodeur génère la séquence de mots pour décrire l'image. En se référant à la génération de phrases dans les travaux de TALN (section 2.2.2.2) et à la représentation des images dans la vision par ordinateur (section 2.2.1), les modèles de génération de descriptions sont composés de réseaux de neurones convolutifs pour encoder l'image et de réseaux de neurones récurrents comme modèle de langue pour décoder.

En considérant l'image I , on définit la loi de probabilité à estimer : $P(w_{1:L}|I)$ dont $w_{1:L}$: la séquence de mots de longueur L . Pour la génération de la séquence, le modèle de langue estime la loi de probabilité $P(w_i|w_1 : i - 1, I)$ qui représente la probabilité de générer un mot w_i sachant la séquence de mots observée $w_{1:i-1}$ et l'image I .

La figure [figure 2.6] illustre l'architecture générale des modèles de génération de descriptions d'une image.

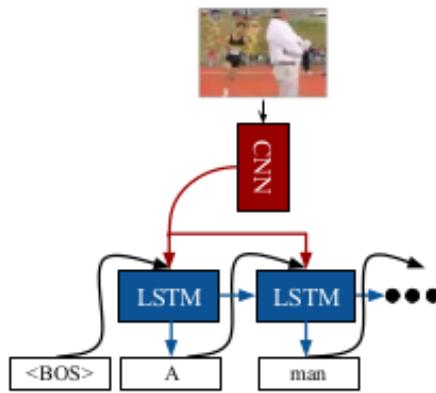


FIGURE 2.6 – Illustration du modèle CNN-LSTM pour la génération de phrases décrivant l'image [8]

Le réseau de neurones convolutifs fournit la caractéristique visuelle de l'image tandis que le modèle de langue est entraîné pour prédire chaque mot de la phrase descriptive donnée par la probabilité $P(w_i|w_1 : i - 1, I)$.

La principale contribution de ces modèles de description d'images s'intéresse à la manière de combiner l'information visuelle (issus du réseau de neurones convolutifs), textuelle (grâce aux méthodes de "word embedding") et le contexte (stocké par le réseau de neurones récurrents) pour générer la séquence de mots qui décrit l'image.

LRCN [8] est un modèle CNN-LSTM créé pour les tâches de la vision par ordinateur impliquant un traitement de séquences de données : reconnaissance d'activité, description d'une image et vidéos. Pour la génération de descriptions d'images LRCN propose 3 variations ($LRCN_{1u}$, $LRCN_{2u}$, $LRCN_{2f}$) de son modèle se basant sur le nombre de couches LSTM et l'introduction de l'information visuelle. $LRCN_{1u}$ et $LRCN_{2u}$ sont composés respectivement d'une seule couche et de deux couches de LSTM. Dans ces deux modèles, le vecteur caractéristique visuel est concaténé avec la représentation textuelle et imbriqué dans le premier LSTM de l'empilement. Tandis que pour $LRCN_{2f}$ (composé de deux couches) le vecteur caractéristique visuelle est concaténé avec l'état caché précédent avant d'être introduit dans le LSTM de la couche courante. Avec les mêmes configurations, $LRCN_{2f}$ a obtenu la meilleure performance sur les trois modèles pour son architecture.

Dans [30] [71], le modèle de génération de descriptions est basé sur un RNN multimodal. Le réseau de neurones récurrents génère une séquence de mots en relation avec l'image en initialisant sa couche cachée par le vecteur caractéristique visuel. Le RNN prend en entrée le vecteur caractéristique visuel une seule fois à l'instant $t = 1$ pour [30] et $t = -1$ dans [71]. Le RNN multimodal est définie par :

$$b_v = W_{hi}[CNN_{\theta_c}(I)] \quad (2.19)$$

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1} + b_h + \mathbb{1}(t = i) \odot b_v) \quad (2.20)$$

$$y_t = softmax(W_{oh}h_t + b_o). \quad (2.21)$$

$CNN_{\theta_c}(I)$: est le vecteur caractéristique de l'image I issus de la dernière couche d'un réseau de neurones convolutifs.

x_t : est le vecteur représentatif du mot à l'instant t .

$\mathbb{1}$ est une fonction indicatrice

$\mathbb{1}(t = i)$: indique l'instant auquel le vecteur caractéristique de l'image I représenté par b_v est injecté.

$i = -1$ pour [71]

$i = 1$ pour [30]

Chapitre 3

Contribution au modèle théorique

Afin d'améliorer le modèle de génération de descriptions d'images proposé par Mao et al. dans [42] et [41] et présenté dans ce rapport en section 3.1, nous avons introduit dans ce dernier plus d'informations sur l'image. Ces informations sont représentées par un vecteur, nommé "vecteur de catégories". Ce vecteur représente les catégories (prédéfinies dans la collection de données) détectées et non détectées sous forme de scores.

3.1 Présentation de la contribution

Inspirée de [26] [79], notre objectif est d'intégrer plus d'informations sémantiques, dans les modèles de génération de descriptions pour les améliorer. [26] propose une extension du LSTM nommée gLSTM (pour guided LSTM). A la différence du LSTM, gLSTM prend des informations sémantiques issues de l'image comme entrées supplémentaires. Dans les modèles LSTM [42] [71] [8] ; [26] Jia et al. ont affirmé que les phrases générées dévient du (ne correspondent pas au) contenu de l'image à cause du comportement instable du décodeur : d'une part la phrase générée doit décrire le contenu de l'image et d'une autre elle doit être un modèle de langue qui définit la séquence de mots la plus envisageable (suivant le contexte et les règles grammaticales). Pour insister sur le contenu de l'image, Dans [79] Xu et al. ont introduit un mécanisme d'attention visuelle. Dans [79], le mécanisme d'attention est représenté par un contexte qui se réfère à une information visuelle capturée sur des régions de l'image. Ainsi les unités du décodeur accordent plus d'attention sur des régions particulières de l'image pour générer la séquence de mots. Le contexte est un vecteur calculé à partir de deux types de mécanisme d'attention : stochastique et déterministe¹. [26] ont utilisé des informations sémantiques de l'image pour guider le modèle de langue. Leur contribution est une extension du LSTM : gLSTM qui, par addition, prend en entrée l'information sémantique de l'image pour orienter le décodeur à favoriser les mots qui sont liés au contenu de l'image. Concrètement, l'information sémantique est ajoutée en entrée dans les unités (*gates*) du LSTM.
Le gLSTM redéfinie les unités de LSTM par :

1. l'explication n'est pas à la portée de ce rapport voir [79]

$$\begin{aligned}
 i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + V^{(i)}g) && \text{input gate} \\
 f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + V^{(f)}g) && \text{forget gate} \\
 o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + V^{(o)}g) && \text{output gate} \\
 \check{c}_t &= \tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) && \text{new memory cell} \\
 c_t &= f_t \circ c_{t-1} + i_t \circ \check{c}_t && \text{final memory cell} \\
 h_t &= o_t \circ \tanh(c_t) && \text{hidden state}
 \end{aligned}$$

g est le vecteur représentatif de l'information sémantique.

Nous proposons une autre alternative pour une extension des modèles de base présentée dans les paragraphes suivants.

Dans notre travail, l'information sémantique d'une image est représentée par un vecteur de scores des catégories : vecteur de catégories. Le vecteur de catégories V_{cat} [figure 3.1] est composé de scores attribués à chaque catégorie pour une image. Cette information permet de guider le modèle à former des phrases plus adaptées au contenu de l'image. Cette hypothèse est justifiée par le fait que la description d'une image est surtout générée à partir des catégories auxquelles elle appartient (exemple [figure 3.1]).

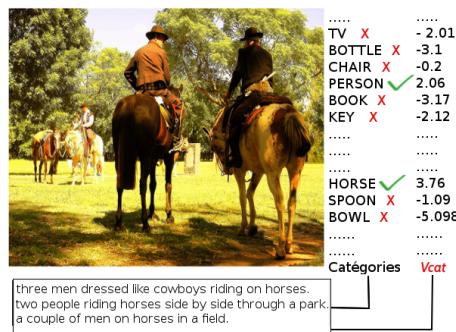


FIGURE 3.1 – Illustration de la relation entre catégories, V_{cat} et descriptions

Les descriptions se rapportent aux catégories auxquelles l'image appartient (person, horse). Notre modèle de classification est entraîné pour prédire ces catégories et leur assigne des scores positifs et pour les autres catégories des scores négatifs. Ces scores forment le vecteur V_{cat} .

Ainsi, on peut redéfinir l'estimation de la loi de probabilité de la séquence sachant la caractéristique visuelle de l'image I : $V(I)$ (issu du réseau de neurones convolutifs) et l'information $V_{cat}(I)$ lors de la classification des images :

$$P(w_i | W_{1:i-1}, V(I), V_{cat}(I)) \quad (3.1)$$

L'équation 3.1 représente la probabilité de générer un mot w_i .

Pour cela, les travaux que nous avons effectués comportent deux parties :

- classification : détaillée dans la section 3.1.1, est la partie dans laquelle les réseaux de neurones convolutifs sont utilisés pour classifier les images dans des catégories prédéfinies dans les données d'apprentissage. Le modèle est utilisé pour prédire les catégories auxquelles une image en entrée appartient. Ce modèle attribue à chaque catégorie un score qui va former le vecteur de catégories.
- génération de descriptions : détaillée dans la section 3.1.2, utilise les résultats de la classification pour apporter l'information sur l'image. Le vecteur de catégories est utilisé comme entrée additionnelle dans notre modèle de base pour estimer la loi de probabilité (3.1). Le modèle de base de notre contribution est un réseau de neurones récurrents multimodal de Juan Mao et al. dans [43] [42] et [41]

3.1.1 Classification des images

Les méthodes de classification ne prennent pas en compte la signification des textes associés aux images. Leur objectif est de classer les images selon les catégories prédéfinies considérées.

Dans les travaux cités sur la reconnaissance de formes et d'objets 2.2.1, les modèles sont entraînés pour prédire une seule catégorie pour une image donnée. Dans notre cas, une image peut appartenir à plusieurs catégories à la fois. Par exemple, l'image d'un chien et de son maître appartient à la catégorie chien et personne à la fois. La classification multi-labels est alors utilisée pour permettre une association multiple entre une image et les catégories. Pour une classification multi-labels, le modèle est entraîné pour prédire les catégories auxquelles une image appartient par estimation de la fonction : $f : X \rightarrow 2^Y$ pour une donnée d'apprentissage : $(x_1, Y_1), (x_2, Y_2), \dots, (x_m, Y_m)$. $x_i \in X$ est une instance de l'i-ème image en entrée représentée par le vecteur descripteur. $Y_i = y_{i1}, \dots, y_{il_i} \subset Y$ est l'ensemble des labels ou catégories associés à l'i-ème image . l_i est le nombre de catégories associées à cette image.

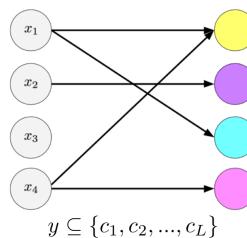


FIGURE 3.2 – Classification multi-labels

Notre objectif est d'extraire le vecteur de catégories composé de score de chaque catégorie pour une image donnée. Notre modèle a été entraîné pour assigner des scores positifs pour les catégories auxquelles l'image appartient et négatifs aux autres. L'apprentissage a été effectué par réglage fin d'un réseau de neurones convolutifs pré-entraîné en optimisant la fonction objectif entropie croisée (équation 3.2).

$$E = \frac{-1}{n} \sum_{n=1}^N [p_n \log \hat{p}_n + (1 - p_n) \log(1 - \hat{p}_n)]. \quad (3.2)$$

$$\hat{p}_n = \sigma(x_n) \in [0, 1]$$

σ est la fonction sigmoïde

x_n est le score prédit par le modèle pour la catégorie n

$p_n = 1$ si l'image appartient à la catégorie n et 0 sinon

Le vecteur de catégories est extrait par propagation directe de l'image sur notre modèle. On définit pour une image I : $V_{cat}(I) = f_{multilabel}(I)$.

$V_{cat}(I) \in \mathbb{R}^C$. C est le nombre de catégories prédéfinies de la collection de données.

3.1.2 Génération de descriptions utilisant le vecteur de catégories

Notre contribution a été intégrée dans le modèle de base [43] [42] nommé m-RNN pour *multimodal Recurrent Neural Network* en anglais. L'architecture de ce modèle nous permet d'expérimenter l'efficacité de l'utilisation du vecteur de catégories pendant l'apprentissage d'un modèle de génération de descriptions.

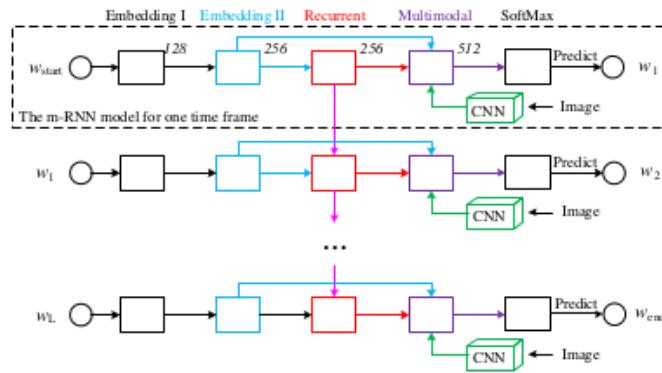


FIGURE 3.3 – Illustration du modèle de base m-RNN [42]

Le modèle m-RNN est composé :

- d'un modèle de langue pour la représentation des mots et phrases
- d'un composant visuel pour la représentation des images
- d'un composant multimodal pour combiner les différentes représentations des informations

Dans le modèle de langue, chaque phrase est représentée par un sac-de-mots binaire selon l'index du mot dans le vocabulaire. Deux couches de neurones successives sont

utilisées pour la représentation vectorielle des mots par la méthode de *word embedding*. Cette représentation est ensuite propagée dans un réseau de neurones récurrents et dans le composant multimodal. Le réseau de neurones récurrent (LSTM ou GRU) est utilisé pour stocker le contexte dans l'état caché.

Le composant visuel est un vecteur caractéristique I de l'image issue d'un réseau de neurones convolutifs pré-entraîné (15-ème couche de VGG-16). Le composant multimodal relie ces 3 sorties : du word embedding, du RNN et du composant visuel pour prédire les mots de la séquence sachant l'image.

Pour une formulation mathématique, nous allons adopter la notation suivante :
à un instant t , soient
 $w(t)$: la représentation finale du mot issu de la seconde couche de word embedding,
 $h(t)$: l'activation (sortie) du RNN,
 $f_2(x)$: la fonction ReLu,
 I : le vecteur caractéristique de l'image issu du réseau de neurones convolutifs,
 $m(t)$: l'activation de la couche multimodale.

$$r(t) = f_2(U_r r(t-1) + w(t)) \quad (3.3)$$

$$m(t) = g_2(V_w w(t) + V_r r(t) + V_I I) \quad (3.4)$$

U_r est une matrice de projection de l'activation du RNN à l'instant $t-1$ sur le même espace que $w(t)$.

L'activation de la couche multimodale est obtenue par la somme des projections des 3 sorties dans un même espace : l'espace multimodal.

La sortie du modèle est une couche *softmax* qui produit la probabilité de générer chaque mot du vocabulaire à partir de la couche multimodale.

Le modèle est entraîné par rétropropagation en optimisant la fonction logarithme de la vraisemblance. La fonction objectif du modèle calcule la moyenne de la fonction de vraisemblance logarithmique sur les mots sachant le contexte et l'image correspondant dans les phrases d'apprentissage.

$$C = \frac{1}{N} \sum_{i=1}^{N_S} L_i \log_2 PPL(w_{1:L_i^{(i)}} | I^{(i)}) + \lambda_\theta \|\theta\|_2^2 \quad (3.5)$$

N_S : nombre de phrases de références

N : nombre de mots

L_i : longueur de l'i-ème phrase

$PPL(w_{1:L_i} | I)$: est la perplexité (mesure standard pour les modèles de langue) de la phrase

$w_{1:L}$ sachant l'image I :

$$\log_2 PPL(w_{1:L_i}|I) = -\frac{1}{L} \sum_{n=1}^L \log_2 P(w_n|w_{1:n-1}, I) \quad (3.6)$$

$P(w_n|w_{1:n-1}, I)$ est obtenu sur l'activation de la couche *softmax* et représente la probabilité à générer le mot w_n , sachant I et les mots précédents $w_{1:n-1}$.

Dans notre modèle, on a introduit un autre composant qui apporte au modèle d'origine plus d'information sémantique sur le contenu de l'image. Ce composant projette le vecteur de catégories associé à l'image I : $V_{cat}(I)$ dans l'espace multimodal afin de générer le futur mot de la séquence. Ainsi l'équation 3.4 devient :

$$m(t) = g_2(V_w w(t) + V_r r(t) + V_I I + V_c V_{cat}(I)) \quad (3.7)$$

3.2 Implémentations

Pour la partie classification, nous avons utilisé l'environnement "caffé" [27]. La classification multi-labels est effectuée par réglage fin (ajustement) du modèle pré-entraîné VGG-16 d'Oxford [63]. La couche softmax a été remplacée par une fonction sigmoïde entropie croisée pour l'apprentissage.

Le modèle a été entraîné par descente de gradient stochastique sur des lots de taille 128 et avec un taux d'apprentissage de 0.0001.

Pour la génération de descriptions, notre implémentation s'est basée sur une implémentation sur tensorflow [1] du modèle m-RNN disponible pour tout public². Un modèle pré-entraîné de l'inception V3 a été utilisé comme descripteur pour extraire les vecteurs caractéristiques visuels des images.

Les phrases associées ont été prétraitées comme suit : elles ont été segmentées en séquence de mots. Les mots qui apparaissent moins de cinq fois dans le corpus entier sont filtrés et ne sont pas inclus dans le vocabulaire et sont remplacés par le caractère *< unk >*.

Notre expérimentation concerne 4 modèles.

- Le premier modèle est le modèle de base de notre contribution [42] [41] désigné par "modèle de base".
- Le second modèle "modele-init" est un modèle inspiré par show and tell [71] et deep visual alignment [30] qui consiste à initialiser l'état caché du modèle de langue par le vecteur caractéristique visuel.
- Le troisième modèle est un modèle utilisant des vecteurs de catégories tirés directement à partir des données d'apprentissage. Ce modèle correspond à une version optimale du modèle de notre contribution. Il représente le modèle de notre contribution quand toutes les catégories auxquelles une image appartient sont prédites à 100%.

2. github.com/mjhula/TF-mRNN

TABLE 3.1 – Configuration utilisée par les quatre modèles

taux d'apprentissage	dimension du composant multimodal	dimension de l'encapsulation	nombre de couche RNN	Beam size ³
1.0	2048	1024	1	3

- Le quatrième modèle est notre modèle qui utilise des vecteurs de catégories prédits par le modèle de classification précédente.

Ces modèles utilisent une même configuration donnée par le tableau 3.1.

3. Beam search : Considérer itérativement les k meilleures séquences à l'intant t pour générer les séquences en $t + 1$. k est appelé *beam size*

Chapitre 4

Évaluation de notre proposition

4.1 Ressources expérimentales

Les ressources utilisées lors de notre expérimentation sont similaires à celles utilisées dans le modèle de base [42] [41] pour une meilleure comparaison des résultats et l'estimation de l'amélioration effectuée sur ce modèle.

4.1.1 Collection de données :

La collection de données utilisée est la collection de Microsoft Common Objects in COntext : MS COCO citelin2014microsoft. MS COCO est une des collections les plus utilisées pour l'expérimentation des modèles de génération de descriptions comme Flickr8k [20] Flickr30k [81]. MS COCO contient des images naturelles classées dans 90 catégories. La majorité des images de cette collection sont des images non iconiques qui permettent aux modèles de mieux généraliser lors de leur apprentissage.

Nous avons utilisé la collection MS COCO Captions [6] qui utilisent les images collectées par MS COCO issus de 80 catégories d'objets et de scènes. Dans sa version actuelle, MS COCO Caption contient 82783 images d'apprentissage et 40504 images de validation. Chaque image est annotée par cinq phrases descriptives.

Avec cette collection de données, nous avons généré 13691 mots dans le vocabulaire. Pour l'apprentissage des modèles, nous avons utilisé les 80000 images d'apprentissage de la collection. Pour évaluer les modèles, nous avons extrait aléatoirement sur les images de validation : 4000 images pour la validation et 1000 images pour le test.

4.1.2 Mesures d'évaluation

4.1.2.1 Mesures d'évaluation de la classification :

Pour la classification multi-labels, notre modèle est évalué par quatre mesures. Soient Y_i l'ensemble des labels corrects pour une instance donnée i et \hat{Y}_i l'ensemble des labels prédits par notre modèle pour cette instance.



<http://mscoco.org/explore/?id=44952>
three men dressed like cowboys riding
on horses
two people riding horses side by side
through a park.
there are several men that are riding
horses together
a couple of men on horses in a field.
cowboys on horseback gather on a
grassy field.

[horse] [personne]



<http://mscoco.org/explore/?id=534468>
a man is watching tv while sitting at
a table
a young man is at a counter filled
with the things of work.
a young man watching television at a
desk with a laptop and a notebook.
a man with a remote pointed at a
television screen sitting beside an
open laptop.
a person at a desk with a laptop and
a note book
[tv] [bottle] [chair] [person]
[refrigerator] [cup] [laptop] [remote]
[bowl] [banana] [dining table]
[keyboard] [book] [cell phone]
[spoon]

FIGURE 4.1 – Exemples de données dans Microsoft COCO Captions [6]

L’erreur de Hamming (hamming loss en anglais) mesure, en moyenne, l’erreur commise par le modèle sur la prédiction de chaque label. Elle prend en compte les labels prédis corrects et les labels pertinents non-prédis.

$$\text{hammingloss} = \frac{1}{NL} \sum_{i=1}^N \sum_{j=1}^L [I(j \in \hat{Y}_i \wedge j \notin Y_i) + I(j \notin \hat{Y}_i \wedge j \in Y_i)]. \quad (4.1)$$

La distance de Hamming est aussi utilisée pour compenser l’erreur de Hamming lors de la mesure de la performance du modèle. Elle est donnée par l’équation 4.2.

$$\text{hammingdistance} = \frac{1}{NL} \sum_{i=1}^N \sum_{j=1}^L [I(j \in Y_i \wedge j \in \hat{Y}_i)]. \quad (4.2)$$

L’accuracy pour chaque instance est définie comme la proportion des labels corrects prédis sur le nombre total de labels de cette instance. L’accuracy est la moyenne sur toutes les instances considérées.

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N \frac{\|Y_i \cap \hat{Y}_i\|}{\|Y_i \cup \hat{Y}_i\|} \quad (4.3)$$

La mesure ”exact match ratio” est une mesure qui ne prend pas en compte les prédictions partiellement correctes mais les considère comme incorrectes. Elle est définie par l’equation 4.4.

$$MR = \frac{1}{N} \sum_{i=1}^N I(Y_i = \hat{Y}_i) \quad (4.4)$$

I est la fonction indicatrice. N est le nombre d’instances considérées.

4.1.2.2 Mesures d'évaluation des descriptions générées :

Une description est jugée selon sa signification (elle doit décrire avec précision l'image) et sa syntaxe (elle doit être grammaticalement correcte). Cependant, aucune mesure n'est aujourd'hui appropriée pour l'évaluation automatique de ces critères [81] [53]. L'évaluation automatique des descriptions générées s'inspire donc des mesures utilisées pour évaluer les systèmes de traduction et de génération de résumés. Dans notre cas, les mesures sont les mêmes utilisées dans le concours MS COCO Captions [6].

L'évaluation s'applique sur la qualité des descriptions générées (phrases candidates) par rapport aux descriptions dans la collection (phrases de références) sachant une image donnée. Pour la suite, nous avons adopté la notation citée dans [6] pour la définition des mesures d'évaluation des modèles entraînés sur cette collection [42] [41] [9] [79] [30] [8]. L'évaluation automatique mesure pour une image I_i la description candidate c_i sachant un ensemble de descriptions de références $S_i = \{s_{i1}, \dots, s_{im}\}$ appartenant à S . Les phrases descriptives sont représentées par un ensemble de n-grammes. $h_k(s_{ij})$ est le nombre d'occurrence d'un n-gramme w_k dans une phrase s_{ij} . $h_k(c_i)$ est le nombre d'occurrence d'un n-gramme w_k dans les candidats c_i appartenant à C

BLEU [54] : BLEU est utilisée pour l'évaluation des systèmes de traduction. Ce type de mesure analyse la cooccurrence des n-grammes entre les phrases générées C et les phrases du corpus de référence S . Soit la précision entre les phrases coupées en n-gram $CP_n(C, S)$

$$CP_n(C, S) = \frac{\sum_i \sum_k \min(h_k(c_i), \max_{j \in m} h_k(s_{ij}))}{\sum_i \sum_k h_k(c_i)} \quad (4.5)$$

k : index de l'ensemble des n-gram possibles

Cette précision $CP_n(C, S)$ est accompagnée d'une *pénalité de brièveté* de la phrase $b(C, S)$ du fait qu'elle favorise les phrases courtes [54].

$$b(C, S) = \begin{cases} 1 & \text{si } l_C > l_S \\ e^{1-l_S/l_C} & \text{si } l_C \leq l_S \end{cases}$$

l_C et l_S sont les longueurs totales respectives des phrases candidates c_i et du corpus de référence.

La mesure BLEU est calculée par :

$$BLEUN(C, S) = b(C, S) \exp \left(\sum_{n=1}^N w_n \log CP_n(C, S) \right) \quad (4.6)$$

$N = 1, 2, 3, 4$ (BLEU-1, BLEU-2, BLEU-3, BLEU-4)

Selon [6], BLEU a montré de bonnes performances pour les comparaisons au niveau du corpus sur lequel un grand nombre de n-gram sont identiques. Cependant pour une comparaison (individuelle) entre les n-grammes des phrases, les correspondances ne se produisent que rarement. Ainsi BLEU n'est pas très appropriée pour la comparaison phrase à phrase. Cette mesure est utilisée dans la génération de description pour comparer les descriptions

générées (phrases candidates) par le modèle et les descriptions issues de la collection de données (corpus de référence).

METEOR [7] : Un alignement est effectué entre les mots de la phrase candidate et de la phrase de référence. L'alignement est calculé en réduisant le nombre de morceaux *ch* (chunk) contigus et identiquement ordonnées des deux phrases. Soit m un ensemble d'alignements, METEOR est une moyenne harmonique de la précision P_m et le rappel R_m entre les phrases de référence et les phrases candidates. Le score final inclut une pénalisation Pen calculée sur le nombre de mots qui se correspondent m et le nombre de morceaux *ch* pour prendre en compte l'ordre des mots des séquences générées. donné par :

$$Pen = \gamma \left(\frac{ch}{m} \right)^\theta \quad (4.7)$$

$$R_m = \frac{|m|}{\sum_k h_k(s_{ij})} \quad (4.8)$$

$$P_m = \frac{|m|}{\sum_k h_k(c_i)} \quad (4.9)$$

$$F_{mean} = \frac{P_m R_m}{\alpha P_m + (1 - \alpha) R_m} \quad (4.10)$$

$$METEOR = (1 - Pen) F_{mean} \quad (4.11)$$

CIDEr [70] Cette mesure est propre à l'évaluation des modèles de génération de légendes et est la mesure la plus corrélée avec le jugement humain [72].

CIDEr mesure un consensus dans les descriptions en calculant le TF-IDF (Term Frequency Inverse Document Frequency) pondéré pour chaque n-gramme. CIDEr calcule le TF-IDF pondéré pour chaque n-gramme par :

$$g_k(s_{ij}) = \frac{h_k(s_{ij})}{\sum_{\omega_l \in \Omega} h_t(s_{ij})} \log \left(\frac{|I|}{\sum_{I_p \in I} \min \left(1, \sum_q h_k(s_{pq}) \right)} \right) \quad (4.12)$$

Ω est le vocabulaire pour les n-grams

I est l'ensemble des images

Le premier terme mesure le TF pour chaque n-gram w_k et le second terme calcule l>IDF.

Pour un n-gram, $CIDEr_n$ est obtenu par une moyenne de la distance entre la phrase candidate et les phrases références (estimé par le rappel et la précision)

$$CIDEr_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{g^n(c_i) \cdot g^n(s_{ij})}{\|g^n(c_i)\| \|g^n(s_{ij})\|} \quad (4.13)$$

$g^n(u)$ est un vecteur formé par les $g^k(u)$ correspondant aux n-grams.

$\|g^n(u)\|$ est la norme du vecteur $g^n(u)$.

$u : c_i$ ou s_{ij} .

En sommant sur les n-gram :

$$CIDEr(c_i, S_i) = \sum_{n=1}^N w_n CIDEr(c_i, S_i) \quad (4.14)$$

$w_n = 1/N$ avec $N = 4$ dans notre cas.

Ces mesures d'évaluation ont été utilisées dans le concours organisé par MS COCO Captioning¹; nous les utilisons également dans notre évaluation.

4.2 Résultats

4.2.1 résultats de la classification

Pendant l'apprentissage du modèle de classification, nous avons suivi son évolution en visualisant l'erreur commise et la qualité de la prédiction mesurée par l'erreur de Hamming, la distance de Hamming et l'*accuracy* (figure 4.2). Selon ces résultats, le modèle commence à être saturé à partir de la 80000 ème itération, point auquel nous avons sauvegardé notre modèle final. A partir du modèle obtenu, nous avons pu extraire le vecteur de catégories associé à une image donnée par propagation directe de cette image (redimensionnée à 224×224) sur ce modèle. A partir de la mesure *exact match ratio* en moyenne, le modèle a pu prédire exactement toutes les catégories pour les 29,4% des images de validation. La distance de Hamming et l'*accuracy* correspondant est à 98% qui mesure la performance partielle du modèle final. Les figures 4.3(a), 4.3(b) et 4.3(c) nous montrent des exemples d'images parfaitement classées, partiellement classées et non-classées par le modèle.

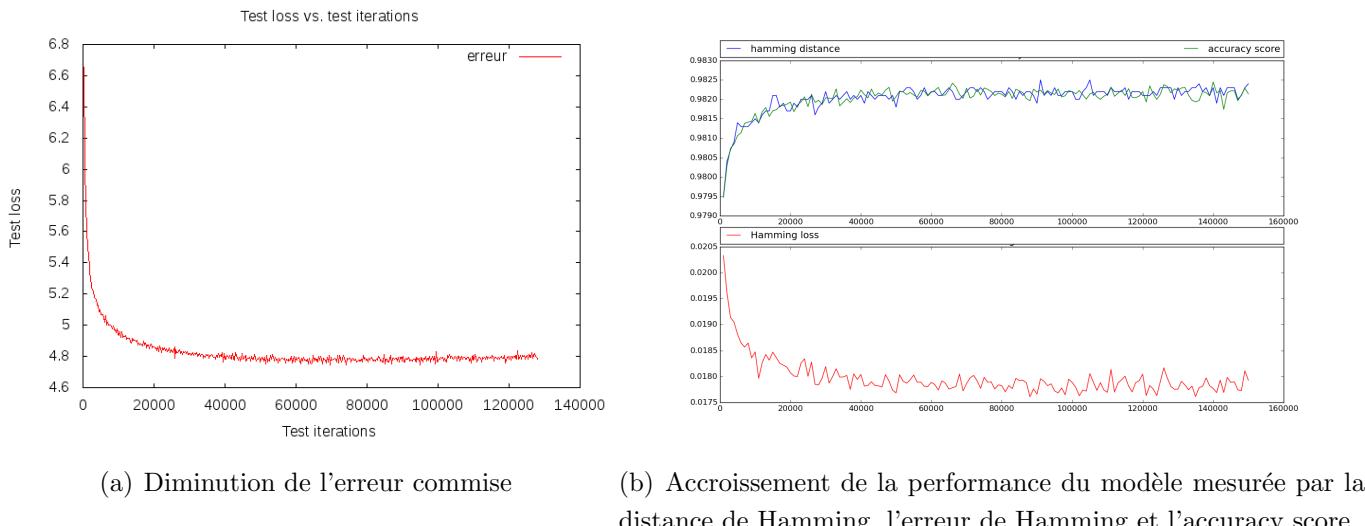


FIGURE 4.2 – Évolution de la performance du modèle de classification pendant l'apprentissage

1. <http://mscoco.org/dataset/#captions-challenge2015> <http://mscoco.org/dataset/#captions-eval>



FIGURE 4.3 – Exemples d’images classées par le modèle

4.2.2 résultats de la génération de descriptions

Les modèles de génération de descriptions ont été comparés au fur et à mesure de leur apprentissage. Pour chaque modèle, les mesures de performance : BLEU-1, BLEU-4, METEOR et CIDEr ont été calculés à chaque 10000 itérations sur les images de test pour évaluer l’évolution des modèles.

D’après les résultats, illustrés par la figure 4.4, on remarque que pour toutes les mesures les courbes de notre modèle (en rouge) sont constamment au-dessus des courbes du modèle de base (en vert). Ceci confirme l’amélioration apportée par notre contribution sur le modèle de base.

Notre modèle a une performance sensiblement similaire à celle du ”model-init” (courbes en bleu) après 40000 itérations.

Les courbes jaunes représentent la performance du modèle optimal de notre contribution et on remarque une nette amélioration de la performance dans ce cas.

Le tableau 4.2.2 montre des exemples de phrases générées à partir de notre modèle. Nous avons pris les meilleures phrases générées par notre modèle et les phrases de la collection similaire à ces dernières.

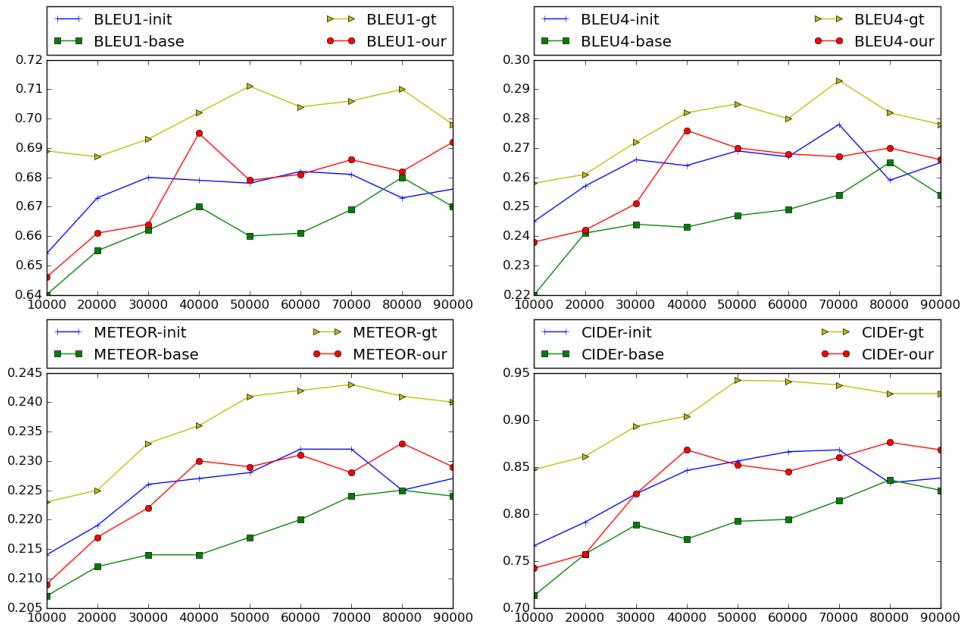
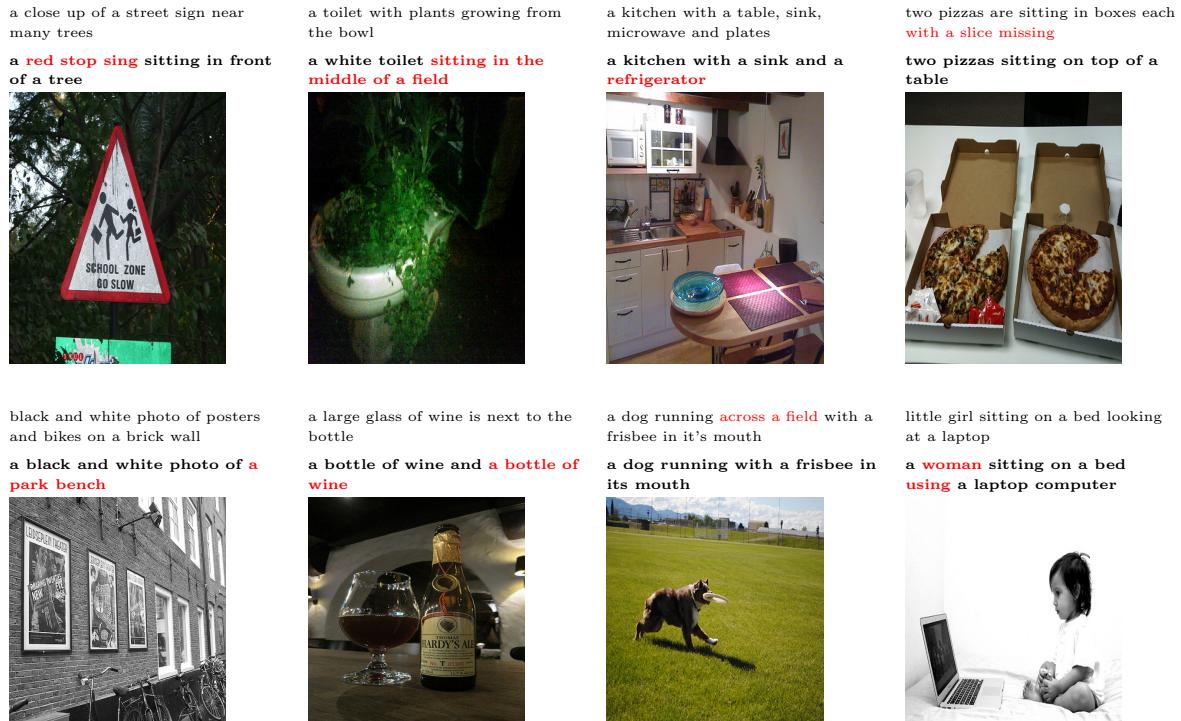


FIGURE 4.4 – Evolution des mesures BLEU-1, BLEU-4, METEOR et CIDEr lors de l'apprentissage des quatre modèles

Pour toutes les mesures, les modèles ayant des scores élevés sont les meilleurs



Les descriptions en gras sont les descriptions générées à partir de notre modèle.

Les mots ou suites de mots en rouge indiquent les erreurs ou les concepts manquants dans les descriptions générées.

TABLE 4.1 – Exemples de descriptions générées

Chapitre 5

Conclusions

5.1 Discussions

D'après les résultats obtenus, l'ajout d'informations sémantiques sur le contenu de l'image améliore la performance du modèle de base m-RNN à générer les phrases descriptives. Ces informations, sous forme de vecteurs de catégories, peuvent être introduites dans le modèle m-RNN par projection sur l'espace multimodal. Une comparaison entre les phrases générées par notre modèle et celles de la collection de données est effectuée pour une évaluation concrète des résultats.

Dans les exemples de descriptions de la table 4.2.2, nous pouvons relever des erreurs qui causent la non-correspondance entre les phrases de la collection des données et les phrases générées par notre modèle. En général, les erreurs concernent la non-identification de certains concepts présents dans l'image. Ces erreurs ont été induites par la performance de la partie visuelle du modèle : le descripteur et notre modèle de classification.

Nous pouvons encore explorer plusieurs idées pour améliorer notre modèle, à savoir :

- une expérimentation sur les hyperparamètres : nous pourrions varier la configuration de notre modèle notamment le nombre de couches du RNN et la dimension de l'espace de projection multimodale pour obtenir un modèle plus performant,
- une amélioration du modèle de classification : en faisant un traitement sur les régions pertinentes de l'image pour atteindre la performance de la version optimale.

5.2 Conclusion et perspective

Le but de notre étude est d'explorer les modèles de l'apprentissage profond pour l'annotation automatique des images. Ainsi, l'analyse des images et textes de la collection de données nous a amené aux travaux issus de la vision par ordinateur et le traitement automatique du langage naturel.

D'une part, les travaux en vision par ordinateur concernent l'extraction de vecteurs caractéristiques des images numériques à partir des réseaux de neurones convolutifs et la classification de ces images.

D'une autre part, les travaux en traitement automatique du langage naturel nous

ont permis de modéliser la génération de séquences de mots en utilisant les réseaux de neurones récurrents.

En associant ces travaux nous avons défini un modèle qui permet de générer automatiquement des phrases décrivant les images .

Nous avons apporté une amélioration du modèle de base multimodal recurrent neural networks : m-RNN [43] [42] pour la description des images de la collection de données de Microsoft COCO Caption [6] en utilisant les informations sur les catégories auxquelles les images appartiennent sous forme de vecteurs de catégories.

Les résultats confirment l'amélioration effectuée sur ce modèle de base sur la performance des modèles par rapport aux mesures BLEU, METEOR et CIDEr.

Ces résultats peuvent encore être améliorés grâce aux travaux futurs proposés dans la discussion pour atteindre la performance du modèle optimal.

Concernant les futurs projets pour l'annotation automatique des images, nous pouvons entamer une annotation des images par génération de descriptions sur des régions de l'image. Cela permet de fournir des informations plus précises et détaillées sur le contenu de l'image en faisant un traitement sur chaque région pertinente de l'image. Visual genome [34], utilisée dans [29], est une collection de données qui permet d'expérimenter sur ce problème.

Annexe

Les RNA pour l'apprentissage profond

Cette section est dédiée à une présentation générale des Réseaux de Neurones Artificiels (RNA) et de l'apprentissage profond [52]. Nous ne présentons que les méthodes les plus utilisées dans les travaux étudiés.

Un réseau de neurones artificiels est un modèle connexionniste qui utilise les informations numériques pour effectuer des calculs analogues à ceux d'un neurone. Les modèles neuronaux imitent la biologie et reproduisent les mécanismes de base naturels. Les réseaux sont constitués de couches successives constituées de neurones. Ces couches sont interconnectées à partir de ces neurones : les neurones de la l-ème couche, qui sont activés, envoient des données aux neurones de la couche suivante par des connections pondérées qui à leur tour calculent leur valeur d'activation ou sortie et la diffusent. Pour une explication plus formelle, nous allons voir l'architecture générale d'un RNA et établir les différentes expressions des traitements effectués par le réseau. En général, un réseau de neurones est composé d'une couche d'entrée, d'une ou plusieurs couches cachées et d'une couche de sortie.

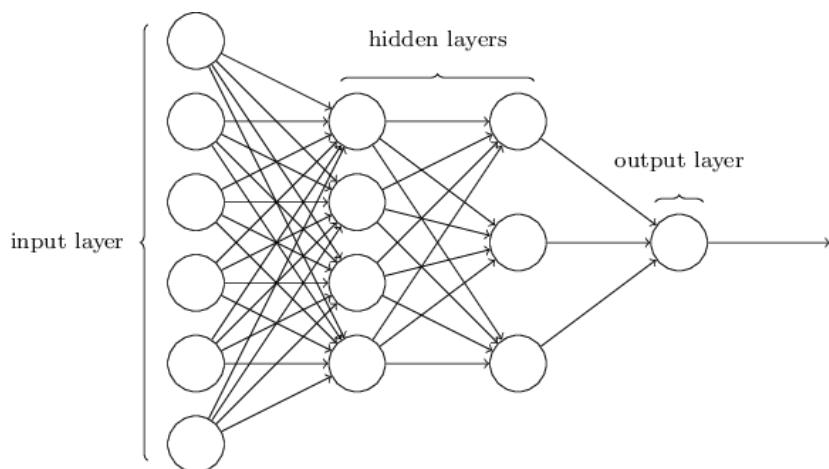


FIGURE 5.1 – Architecture générale d'un réseau de neurones artificiels

La couche d'entrée est composée de neurones qui correspondent aux caractéristiques des données d'entrée représentées par une grille multidimensionnelle (par exemple la matrice de pixels de l'image). La couche de sortie représente les résultats de la tâche assignée

au réseau. Par exemple pour une classification de 1000 classes, les 1000 neurones de la couche de sortie représentent la probabilité pour chaque classe.

Propagation directe :

La propagation directe est le traitement des données d'entrée du réseau jusqu'au calcul des sorties. Ainsi le traitement des données d'entrée se propage de couche en couche. Pour chaque neurone d'une couche l , les entrées (activations des neurones voisins) issus des connexions de ses neurones voisins sont sommées par rapport au poids de chaque connexion pour calculer sa valeur d'activation à partir d'une fonction non-linéaire appelée : fonction d'activation. Pour le j -ème neurone du l -ème couche, on définit sa valeur d'activation a_j^l par rapport à aux sorties de la $(l-1)$ -ème couche des neurones voisins¹ :

$$a_j^l = \sigma \left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l \right) \quad (5.1)$$

b_j^l : est le biais qui contrôle la somme sur les sorties de la $(l-1)$ -ème couche.

w_{jk} : poids de la connexion

σ : est la fonction d'activation des neurones de la l -ème couche.

D'une manière générale, on peut définir pour une couche l :

$$a^l = \sigma(w^l a^{l-1} + b^l) \quad (5.2)$$

a^l : sorties des neurones l -ème couche w^l : matrice des poids de connexions b^l : vecteur des biais

Les fonctions d'activation non-linéaires permettent de contrôler le comportement du modèle à partir des fonctions non-linéaires.

- sigmoid : ou fonction logistique

$$\sigma(z) \equiv \frac{1}{1 + e^{-z}} \quad (5.3)$$

- tangente hyperbolique
- Rectified Linear Units (ReLUs) : $f(x) = \max(0, x)$ qui est souvent utilisé pour une représentation de la probabilité pour de sortie (probabilité qui correspond à l' i -ème classe).

Pour un réseau de neurones composé de L couches, la propagation directe est donnée par la composition de fonctions :

$$f_\theta() = f_\theta^L(f_\theta^{L-1}(\dots(f_\theta^1())\dots)) = f_\theta^L \circ f_\theta^{L-1} \circ \dots \circ f_\theta^1() \quad (5.4)$$

1. Il est à préciser que les neurones d'une couche donnée ne sont pas toujours connectés à tous les neurones de la couche adjacente. Les réseaux de neurones convolutifs utilisent d'autres formes de connection plus complexe. Si c'est le cas, on parle de couches interconnectées

θ : est l'ensemble des paramètres $[w, b]$

La sortie d'un réseau est calculée par propagation directe des entrées. Cette sortie correspond à la solution proposée ou prédite par le modèle de la tâche. L'apprentissage d'un RNA consiste à trouver l'ensemble des paramètres optimaux pour maximiser la performance du modèle à effectuer cette tâche. La recherche de cet ensemble est souvent effectuée par l'optimisation d'une fonction qui mesure l'erreur commise par le modèle appelée : fonction objectif ou coût. La fonction objectif calcule l'erreur à partir de la solution prédite par le modèle et la solution désirée issues des données d'apprentissage (exemple : MSE : Mean Squared Error, log-vraisemblance, entropie croisé). Ainsi le choix de cette fonction est important pour une bonne performance du modèle.

L'approche que nous avons étudiée concerne la rétropropagation de l'erreur pour l'apprentissage d'un RNA. Cette approche est largement utilisée du fait qu'elle est compatible à plusieurs types de sorties de réseau et de fonctions objectifs.

Rétropropagation :

L'algorithme de rétropropagation permet de propager l'erreur calculée à partir de la fonction objectif vers le reste du réseau pour mettre à jour les paramètres du modèle. La rétropropagation est une méthode pour calculer le gradient de la fonction objectif par rapport aux paramètres w le poids et b biais du modèle. Elle est basée par l'application récursive de la règle de la dérivation des fonctions composées ou la règle de la chaîne [équation 5.5].

$$\frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx} \quad (5.5)$$

L'objectif est de calculer le gradient de la fonction objectif C : $\Delta_\theta C = \frac{\partial C}{\partial w}, \frac{\partial C}{\partial b}$. Pour la minimisation on procède à la descente du gradient [équation 5.6] contrôlée par le taux d'apprentissage η qui est un hyperparamètre du modèle.

$$w_k \rightarrow w'_k = w_k - \eta \frac{\partial C}{\partial w_k} \quad (5.6)$$

$$b_l \rightarrow b'_l = b_l - \eta \frac{\partial C}{\partial b_l}. \quad (5.7)$$

Dans le domaine de l'apprentissage profond, la fonction à optimiser est non-convexe rendant la convergence de la descente de gradient difficile (à cause des minimums locaux et points saillants)

En général, l'algorithme pour l'apprentissage de RNA est une itération de propagation directe et de rétropropagation sur les données d'apprentissage [5.2].

Des variations de la descente de gradient peuvent être utilisées pour optimiser le traitement de la minimisation de la fonction objectif. Dans nos travaux, nous avons utilisé la descente de gradient stochastique [équation 5.8] (SGD) pour accélérer le traitement. En résumé, la descente de gradient stochastique calcule le gradient sur plusieurs données

Algorithm 1 apprentissage RNA

Entrée : D données d'apprentissage

while $x \in D$ **do**

 assigner $a^{x,1}$

end while

propagation directe :

for $l = [2, 3, \dots, L]$ **do**

$z^{x,l} = w^l a^{x,l-1} + b^l$

$a^{x,l} = \sigma(z^{x,l})$

end for

erreur de la sortie $\delta^{x,L}$

$\delta^{x,L} = \nabla_a C_x \odot \sigma'(z^{x,L})$

rétropropagation de l'erreur :

for $l = [L-1, L-2, \dots, 2]$ **do**

$\delta^{x,l} = ((w^{l+1})^T \delta^{x,l+1}) \odot \sigma'(z^{x,l})$

end for

descente de gradient :

for $l = L, L-1, \dots, 2$ **do**

 # Mettre à jour les paramètres

$w^l \rightarrow w^l - \frac{\eta}{m} \sum_x \delta^{x,l} (a^{x,l-1})^T$

$b^l \rightarrow b^l - \frac{\eta}{m} \sum_x \delta^{x,l}$

end for

tirées aléatoirement sous forme de lots : mini-batch (traitement par lots)², en même temps et calcule la moyenne pour estimer le gradient de la fonction objectif.

$$w_k \rightarrow w'_k = w_k - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial w_k} \quad (5.8)$$

$$b_l \rightarrow b'_l = b_l - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial b_l}, \quad (5.9)$$

La somme est effectuée sur les données X_j du mini-batch courant

Cette technique peut être combinée par une descente de gradient basée sur le *momentum* β , un hyperparamètre qui permet d'accélérer la descente en optant pour la règle [equation 5.10]

$$v \rightarrow v' = \beta v - \eta \nabla C \quad (5.10)$$

$$w \rightarrow w' = w + v'. \quad (5.11)$$

v correspond à la vélocité du poids w

Généralisation et régularisation :

La généralisation est la capacité du modèle à représenter les nouvelles données. En effet, l'approximation universelle implique que le modèle soit capable de représenter toute donnée lors de l'apprentissage du modèle. Cela n'est pas valable pour les données avec lesquelles le modèle n'a pas été entraîné (les données de test) : ainsi le principe de la généralisation s'impose pour une bonne représentation de ces nouvelles données. Le problème de la généralisation se rapporte à l'analyse de l'erreur commise par le modèle lors de l'apprentissage : *training error* et surtout à l'erreur commise sur les nouvelles données *test error* pour détecter d'éventuelles sous-apprentissage et sur-apprentissage. La généralisation permet aussi au modèle de s'adapter facilement à de larges données et être moins sensible à la dispersion des données. Pour éviter le sur-apprentissage, un terme de régularisation λ est ajouté à la fonction objectif pour encourager les paramètres (poids) à tendre vers zéro et pour tolérer les grandes valeurs que s'ils ont un apport considérable sur l'optimisation de la fonction objectif. Le terme de régularisation ou "*weight decay*" peut être interprété comme un compromis entre l'obtention de faible valeur des poids et la minimisation de la fonction objectif. Les deux types de régularisation sont définis par :

- L2 régularisation :

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2 \quad (5.12)$$

2. la taille du lot est un hyperparamètre du modèle

- L1 régularisation :

$$C = C_0 + \frac{\lambda}{n} \sum_w |w| \quad (5.13)$$

Bibliographie

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow : Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv :1603.04467*, 2016.
- [2] P. Baldi and K. Hornik. Neural networks and principal component analysis : Learning from examples without local minima. *Neural networks*, 2(1) :53–58, 1989.
- [3] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *journal of machine learning research*, 3(Feb) :1137–1155, 2003.
- [4] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank. Automatic description generation from images : A survey of models, datasets, and evaluation measures. *Journal of Artificial Intelligence Research*, 55 :409–442, 2016.
- [5] N. Boujema and M. Ferecatu. Evaluation des systemes de traitement de l’information, chapter evaluation des systemes de recherche par le contenu visuel : pertinence et criteres. *Number ISBN*, pages 2–7462, 2004.
- [6] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft coco captions : Data collection and evaluation server. *arXiv preprint arXiv :1504.00325*, 2015.
- [7] M. Denkowski and A. Lavie. Meteor universal : Language specific translation evaluation for any target language. In *In Proceedings of the Ninth Workshop on Statistical Machine Translation*. Citeseer, 2014.
- [8] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.
- [9] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1482, 2015.

- [10] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise : A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.
- [11] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *International journal of computer vision*, 106(2) :210–233, 2014.
- [12] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *European Conference on Computer Vision*, pages 529–545. Springer, 2014.
- [13] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, volume 14, pages 1764–1772, 2014.
- [14] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis : An overview with application to learning methods. *Neural computation*, 16(12) :2639–2664, 2004.
- [15] Z. S. Harris. Distributional structure. *Word*, 10(2-3) :146–162, 1954.
- [16] R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural Networks, 1989. IJCNN., International Joint Conference on*, pages 593–605. IEEE, 1989.
- [17] N. Hervé. *Vers une description efficace du contenu visuel pour l'annotation automatique d'images*. PhD thesis, Université Paris Sud-Paris XI, 2009.
- [18] S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02) :107–116, 1998.
- [19] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8) :1735–1780, 1997.
- [20] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task : Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47 :853–899, 2013.
- [21] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2333–2338. ACM, 2013.
- [22] Y. B. Ian Goodfellow and A. Courville. Deep learning. Book in preparation for MIT Press, 2016.
- [23] S. Ioffe and C. Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv :1502.03167*, 2015.

- [24] O. Irsøy and C. Cardie. Bidirectional recursive neural networks for token-level labeling with structure. *arXiv preprint arXiv :1312.0493*, 2013.
- [25] K. Janod, M. Morchid, R. Dufour, and G. Lianrès. Réseaux de neurones pour la représentation de contextes continus des mots.
- [26] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars. Guiding long-short term memory for image caption generation. *arXiv preprint arXiv :1509.04942*, 2015.
- [27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe : Convolutional architecture for fast feature embedding. *arXiv preprint arXiv :1408.5093*, 2014.
- [28] R. J. T. JitendraMalik. Rich feature hierarchies for accurate object detection and semantic segmentation.
- [29] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap : Fully convolutional localization networks for dense captioning. *arXiv preprint arXiv :1511.07571*, 2015.
- [30] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.
- [31] A. Karpathy, J. Johnson, and L. Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv :1506.02078*, 2015.
- [32] A. Karpathy, A. Joulin, and F. F. Li. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*, pages 1889–1897, 2014.
- [33] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv :1411.2539*, 2014.
- [34] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome : Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv :1602.07332*, 2016.
- [35] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.
- [37] C.-Y. Lin. Rouge : A package for automatic evaluation of summaries. In *Text summarization branches out : Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.

- [38] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco : Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.
- [39] B. Liu, Y. Liu, and K. Zhou. Image classification for dogs and cats.
- [40] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov) :2579–2605, 2008.
- [41] J. Mao, X. Wei, Y. Yang, J. Wang, Z. Huang, and A. L. Yuille. Learning like a child : Fast novel visual concept learning from sentence descriptions of images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2533–2541, 2015.
- [42] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv :1412.6632*, 2014.
- [43] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv :1410.1090*, 2014.
- [44] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4) :115–133, 1943.
- [45] F. Mikolajczyk. Deep correlation for matching images and text. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 2015.
- [46] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*, 2013.
- [47] T. Mikolov and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.
- [48] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.
- [49] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531. IEEE, 2011.
- [50] T. M. Mitchell. Machine learning. *New York*, 1997.
- [51] V. N. Murthy, S. Maji, and R. Manmatha. Automatic image annotation using deep learning representations. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 603–606. ACM, 2015.
- [52] M. Nielsen. Neural networks and deep learning [consulté le 21 mars 2016], Januar 2016. Disponible sur <https://en.wikipedia.org/>.

- [53] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text : Describing images using 1 million captioned photographs. In *Advances in Neural Information Processing Systems*, pages 1143–1151, 2011.
- [54] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.
- [55] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 251–260. ACM, 2010.
- [56] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn : Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [57] F. Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6) :386, 1958.
- [58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3) :211–252, 2015.
- [59] M. Sahlgren. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1) :33–54, 2008.
- [60] J. Schmidhuber. Deep learning in neural networks : An overview. *Neural Networks*, 61 :85–117, 2015.
- [61] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11) :2673–2681, 1997.
- [62] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf : an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.
- [63] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*, 2014.
- [64] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12) :1349–1380, 2000.
- [65] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

- [66] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [67] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv :1512.00567*, 2015.
- [68] M. Szummer and R. W. Picard. Indoor-outdoor image classification. In *Content-Based Access of Image and Video Database, 1998. Proceedings., 1998 IEEE International Workshop on*, pages 42–51. IEEE, 1998.
- [69] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011.
- [70] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider : Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, 2015.
- [71] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell : A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.
- [72] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell : Lessons learned from the 2015 mscoco image captioning challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.
- [73] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. *arXiv preprint arXiv :1511.06078*, 2015.
- [74] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. Cnn : Single-label to multi-label. *arXiv preprint arXiv :1406.5726*, 2014.
- [75] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb) :207–244, 2009.
- [76] P. J. Werbos. Backpropagation through time : what it does and how to do it. *Proceedings of the IEEE*, 78(10) :1550–1560, 1990.
- [77] Wikipédia. Apprentissage automatique [consulté le 15 avril 2016], May 2016. Disponible sur https://fr.wikipedia.org/wiki/Apprentissage_automatique.
- [78] Wikipédia. Kernel (image processing) [consulté le 27 aout 2016], May 2016. Disponible sur <https://en.wikipedia.org/>.

- [79] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell : Neural image caption generation with visual attention. *arXiv preprint arXiv :1502.03044*, 2(3) :5, 2015.
- [80] F. Yan and K. Mikolajczyk. Deep correlation for matching images and text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3441–3450, 2015.
- [81] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations : New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2 :67–78, 2014.
- [82] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

Résumé

Récemment, les travaux sur l'annotation automatique d'images s'intéressent à la description d'images par des légendes : « *image captionning* » en anglais. Dans ce travail, nous nous sommes focalisés sur les modèles multimodaux pour générer des phrases descriptives pour une image donnée par apprentissage profond. Ces modèles combinent les travaux issus de la vision par ordinateur et le traitement automatique du langage naturel pour analyser images et textes afin de les associer. Notre contribution est une amélioration du modèle m-RNN [43] [42] pour la génération de descriptions de la collection de données de Microsoft COCO Caption [6]. L'amélioration exploite les catégories auxquelles une image appartient pour pouvoir guider le modèle de langue à générer des phrases en relation avec le contenu de cette image.

Mots clés :apprentissage profond, réseau de neurones artificiels, annotation automatique d'images, génération de légendes

Abstract

This thesis tackles the problem of annotating images automatically that corresponds to image captions.

In this work, we focus on multimodal models for generating descriptive sentences for a given image via deep learning models. These models combine computer vision and natural language processing to analyze images and texts. Our contribution improves the multimodal recurrent neural network : m-RNN model [43] [42] for image captioning on the benchmark dataset : Microsoft data collection COCO Caption [6]. We exploited the categories an image belongs to, in order to guide the language model and generate more accurate descriptive sentences.

Keywords : deep learning, artificial neural networks, automatic image annotation, image captioning

Titre : Annotation automatique d'images par apprentissage profond : Génération automatique de descriptions d'une image

Auteur : Nomena Fitiavana NY HOAVY

Tel : +261332056075

Email : nyhoavynomena@yahoo.ca

Encadreur : Madame Josiane MOTHE