جامعة شعيب الدكالي
Université Chouaïb Doukkali

ENSA
EL JADIDA

Rapport de TP : Attention

Génération de légendes d'images avec CNN, RNN et Mécanisme d'Attention

Réalisé par :
Yassine KADER

Encadré par :
Pr. Youness ABOUQORA

Matière :
IA GENERATIVE ET INGENIERIE DES PROMPTS

December 25, 2025

Contents

1 Introduction

L'objectif de ce travail pratique est de développer un modèle de *Image Captioning* (légendage automatique d'images). Cette tâche consiste à générer une description textuelle naturelle et pertinente pour une image donnée. Elle se situe à l'intersection de deux domaines majeurs de l'intelligence artificielle : la Vision par Ordinateur (*Computer Vision*) et le Traitement Automatique du Langage Naturel (*Natural Language Processing* ou NLP).

Le défi réside dans le fait que le modèle doit non seulement reconnaître les objets présents dans l'image (détection d'objets), mais aussi comprendre leurs relations spatiales, leurs actions et leurs attributs, pour ensuite articuler ces informations dans une phrase grammaticalement correcte.

Dans ce TP, nous implémentons une architecture hybride qui combine :

- Un **Encodeur visuel** basé sur un Réseau de Neurones Convolutif (CNN), spécifiquement un ResNet50 pré-entraîné, pour extraire les caractéristiques de l'image.
- Un **Décodeur séquentiel** basé sur un Réseau de Neurones Récurrent (RNN), spécifiquement un LSTM (*Long Short-Term Memory*), pour générer la phrase mot par mot.
- Un **Mécanisme d'Attention** qui permet au décodeur de se focaliser sur des zones spécifiques de l'image à chaque pas de la génération, améliorant ainsi la précision et la pertinence des descriptions.

Nous utilisons le dataset **Flickr30k**, qui contient plus de 30 000 images, chacune annotée avec plusieurs descriptions, offrant un corpus riche et varié pour l'entraînement.

2 Cadre Théorique

2.1 Réseaux de Neurones Convolutifs (CNN) et Transfer Learning

Les CNN sont l'état de l'art pour le traitement des images. Ils utilisent des couches de convolution pour apprendre hiérarchiquement des motifs visuels, allant des bords simples aux formes complexes et aux objets entiers.

Pour ce projet, nous utilisons l'apprentissage par transfert (*Transfer Learning*). Entraîner un CNN profond à partir de zéro nécessite des millions d'images et une puissance de calcul considérable. Au lieu de cela, nous utilisons **ResNet50**, un modèle pré-entraîné sur le dataset ImageNet (1,2 million d'images, 1000 classes).

2.1.1 Architecture ResNet et Apprentissage Résiduel

ResNet (Residual Network) a introduit le concept de connexions résiduelles (*skip connections*). Dans un réseau très profond, le gradient a tendance à s'évanouir (*vanishing gradient*) lors de la rétropropagation, rendant l'apprentissage difficile. Les blocs résiduels permettent au gradient de "sauter" des couches :

$$y = \mathcal{F}(x, \{W_i\}) + x \tag{1}$$

Cela permet d'entraîner des réseaux beaucoup plus profonds (50, 101, ou 152 couches) tout en facilitant l'optimisation.

Dans notre cas, nous supprimons les dernières couches de classification de ResNet50 pour récupérer les cartes de caractéristiques (*feature maps*) brutes. Ces cartes conservent l'information spatiale de l'image, ce qui est crucial pour le mécanisme d'attention.

2.2 Réseaux Récurrents (RNN) et LSTM

Les RNN sont conçus pour traiter des données séquentielles. Ils possèdent une "mémoire" sous forme d'état caché h_t qui est mis à jour à chaque pas de temps t.

Cependant, les RNN classiques souffrent du problème de disparition de gradient sur de longues séquences. Pour pallier cela, nous utilisons un **LSTM** (Long Short-Term Memory). Le LSTM introduit une cellule mémoire c_t et trois portes logiques qui régulent le flux d'information :

- **Porte d'entrée (i_t)** : Décide quelle nouvelle information stocker dans la cellule.
- **Porte d'oubli (f_t)** : Décide quelle information supprimer de la cellule.
- **Porte de sortie (o_t)** : Décide quelle information envoyer en sortie (vers l'état caché h_t).

2.3 Mécanisme d'Attention (Soft Attention)

Dans une approche encodeur-décodeur classique, toute l'image est compressée en un seul vecteur fixe. Cela pose problème pour les images complexes où plusieurs détails sont importants.

L'attention permet au modèle de calculer une "carte de pertinence" à chaque instant. Au lieu de regarder toute l'image avec la même intensité, le décodeur attribue un poids α_{ti} à chaque région i de l'image au temps t.

$$z_t = \sum_i \alpha_{ti} a_i \tag{2}$$

où a_i sont les vecteurs de caractéristiques spatiaux extraits par le CNN et z_t est le vecteur de contexte dynamique. Les poids α sont appris par le réseau lui-même.

3 Implémentation Détaillée

3.1 Préparation des Données et Embeddings

Le dataset Flickr30k a subi un prétraitement rigoureux.

- **Normalisation** : Les images sont centrées réduites (moyenne et écart-type d'ImageNet).
- **Vocabulaire** : Nous avons construit un dictionnaire associant chaque mot unique à un index entier.
- **Word2Vec** : Pour enrichir la représentation sémantique, nous avons initialisé la couche d'embedding avec des vecteurs pré-entraînés (Google News 300d). Cela permet au modèle de comprendre que "chien" et "chiot" sont sémantiquement proches dès le début de l'entraînement.

3.2 Module d'Attention Personnalisé

Au lieu de présenter le code source, nous décrivons ici l'architecture détaillée du module d'attention que nous avons implémenté. Ce module agit comme une interface intelligente entre les caractéristiques visuelles (CNN) et l'état courant du décodeur (LSTM).

Composant	Opération / Dimension	Rôle
Inputs	**Visual Features**: $(Batch, 2048, H, W)$ **Hidden State**: $(Batch, Hidden_{dim})$	Information visuelle brute et état actuel de la génération.
Flattening	Reshape $\rightarrow$ $(Batch, L, 2048)$	Aplatissement des dimensions spatiales ($L = H \times W$).
Expansion	Repeat Hidden $\rightarrow$ $(Batch, L, Hidden_{dim})$	Alignement de l'état caché avec chaque pixel.
Concaténation	Concat $\rightarrow$ $(Batch, L, 2048+ Hidden_{dim})$	Fusion de l'information visuelle et contextuelle.
Linear 1 (W_att)	Linear $\rightarrow$ $(Batch, L, Att_{dim})$ + Tanh	Projection dans l'espace d'attention latent.
Linear 2 (V_att)	Linear $\rightarrow$ $(Batch, L, 1)$	Calcul du score de pertinence brut pour chaque pixel.
Attention Weights	Softmax sur la dim L	Normalisation pour obtenir une distribution de probabilité ($\sum \alpha = 1$).
Context Vector (z_t)	Somme pondérée	Création du vecteur final résumant les zones pertinentes de l'image.

Table 1: Architecture fonctionnelle du module d'Attention.

3.3 LSTM avec Attention Intégrée

Le cœur du décodeur est un LSTM modifié où le vecteur de contexte est injecté à chaque étape de calcul des portes. Voici la structure des opérations par pas de temps :

Étape	Entrées Combinées	Calcul Effectué
Input Combination	x_t (Word Emb), h_{t-1}, z_t (Context)	Concaténation des trois vecteurs sources.
Input Gate (i_t)	Vecteur Combiné	$\sigma(W_i \cdot Combined + b_i)$ - Contrôle l'ajout d'info.
Forget Gate (f_t)	Vecteur Combiné	$\sigma(W_f \cdot Combined + b_f)$ - Contrôle l'oubli.
Output Gate (o_t)	Vecteur Combiné	$\sigma(W_o \cdot Combined + b_o)$ - Contrôle la sortie.
Cell Update (c_t)	f_t, c_{t-1}, i_t	Mise à jour de la mémoire interne.
Output (h_t)	o_t, c_t	$o_t \cdot \tanh(c_t)$ - Nouvel état caché du RNN.

Table 2: Structure interne de la cellule LSTM modifiée.

3.4 Question : Pourquoi geler les poids du ResNet ? (I.6)

Dans ce TP, nous avons gelé (*freeze*) les poids du modèle ResNet50 (en mettant `requires_grad = False`). Cette étape est cruciale pour plusieurs raisons :

1. **Prévention du sur-apprentissage (Overfitting)** : Notre dataset Flickr30k (30k images) est relativement petit comparé à ImageNet (1.2M images). Si nous ré-entraînions tout le ResNet (25M de paramètres), le modèle mémoriserait rapidement les images au lieu d'apprendre des caractéristiques généralisables.
2. **Préservation des connaissances** : ResNet50 a déjà appris des filtres très performants pour détecter des formes, textures et objets. Nous voulons utiliser ces connaissances telles quelles, agissant comme un extracteur de caractéristiques fixe et robuste.
3. **Efficacité computationnelle** : Calculer les gradients pour les 50 couches du ResNet à chaque itération est très coûteux en mémoire (VRAM) et en temps de calcul. Geler les poids accélère considérablement l'entraînement de la partie RNN.

4 Résultats Expérimentaux

4.1 Protocole d'Entraînement

Le modèle a été entraîné avec les hyperparamètres suivants :

- **Optimiseur** : Adam (Adaptive Moment Estimation), reconnu pour sa rapidité de convergence.
- **Fonction de Coût** : CrossEntropyLoss, classique pour les tâches de classification de mots.
- **Learning Rate** : Initialisé à 0.001, avec un *Scheduler* de type `StepLR` qui réduit le taux d'apprentissage périodiquement pour affiner la convergence.
- **Taille de Batch** : 32.

4.2 Analyse de la Convergence

Les graphiques ci-dessous montrent l'évolution de la fonction de perte (Loss) au cours de l'entraînement.

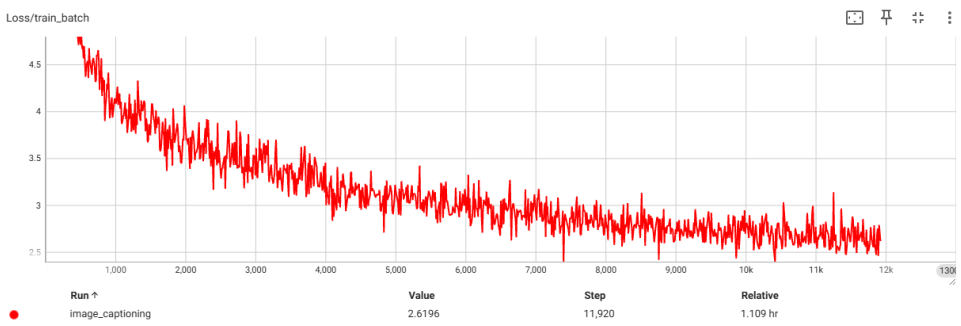

Figure 1: Évolution de la Perte par batch. La courbe montre une diminution bruitée mais constante, typique de la descente de gradient stochastique.

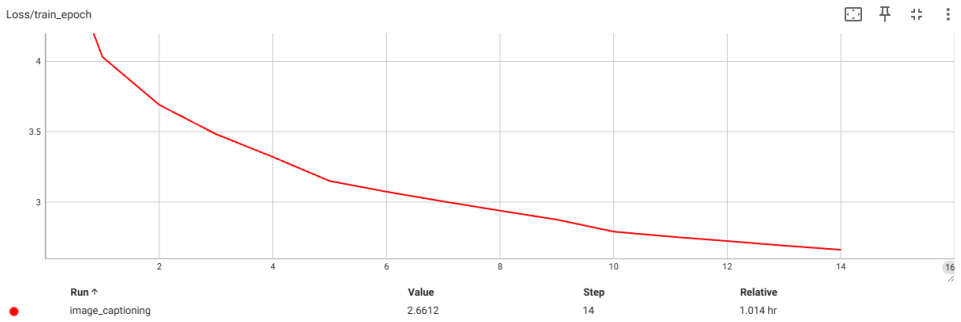

Figure 2: Évolution de la Perte moyenne par époque. La décroissance est fluide, indiquant un apprentissage stable.

Le Scheduler a réduit le Learning Rate par paliers, ce qui est visible sur la courbe ci-dessous. Cette technique permet de faire de grands pas au début de l'optimisation, puis de plus petits pas pour converger vers un minimum local précis.

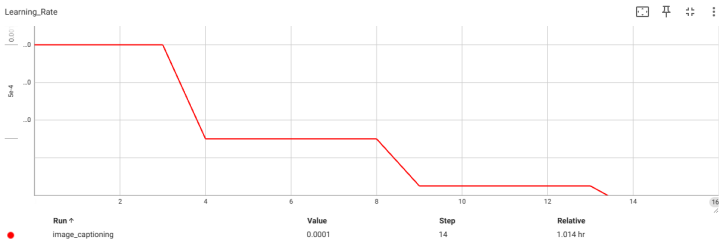

Figure 3: Évolution du Learning Rate (Step Decay).

4.3 Performance Quantitative

Le tableau ci-dessous détaille la progression numérique de la perte. On constate une réduction drastique de l'erreur, divisée par plus de 3 entre le début et la fin.

Itération	**Phase**	**Loss (Train)**
0	Initialisation	9.89
100	Début	5.69
1000	Epoch 1	4.10
3000	Epoch 4	3.47
7000	Epoch 10	3.00
9000	Epoch 13	2.75
11000	Fin	2.60

Table 3: Tableau récapitulatif de la convergence.

4.4 Analyse Qualitative

Nous avons soumis des images de l'ensemble de test (jamais vues par le modèle) pour générer des légendes. Les résultats montrent que le modèle a acquis une compréhension sémantique de la scène. Il est capable d'identifier les acteurs principaux (hommes, femmes, enfants, chiens), leurs actions (courir, jouer, s'asseoir) et souvent le contexte (sur l'herbe, dans la rue).

Bien que certaines erreurs grammaticales ou de détails puissent subsister, la cohérence globale est satisfaisante. Le modèle parvient à lier les objets détectés par le CNN via le mécanisme d'attention pour former une phrase logique.

Generated: a woman in a black shirt and black pants is jumping into the air while a woman in a

Actual: a woman figure skater in a blue costume holds her leg in the air by the blade of her

Generated: a girl in a blue shirt and a blue shirt is standing in a wooded area <UNK>

Actual: a little girl has her arms around a little boy standing on a wooden bridge in the woods <UNK>

Figure 4: Exemples de générations réussies.

Generated: a man in a blue shirt and black pants is playing a ball in a field <UNK>

Actual: a golfer wearing black pants and a black shirt swings at a golf ball lying on short cut green

Generated: a young boy wearing a blue shirt and a woman in a blue shirt is standing in a grassy

Actual: a young south american child sits alone on a dirt and grass covered knoll along a ravine <UNK>

Figure 5: Autres exemples de prédictions.

5 Discussion et Perspectives

5.1 Métriques d'Évaluation Automatisées

Dans ce TP, nous avons principalement utilisé la Loss et l'inspection visuelle. Cependant, pour une évaluation scientifique rigoureuse, il serait nécessaire d'utiliser des métriques standardisées en NLP :

- **BLEU (Bilingual Evaluation Understudy)** : Mesure le chevauchement des n-grams entre la génération et les références. C'est la métrique historique.
- **METEOR** : Prend en compte la synonymie et la racinisation (stemming).
- **CIDEr** : Spécifique à l'image captioning, elle pondère les n-grams par TF-IDF pour donner plus d'importance aux mots rares et informatifs.

5.2 Limites de l'Approche Actuelle

Notre modèle LSTM + Attention présente quelques limitations :

- **Sérialité** : Le LSTM traite les mots un par un, ce qui empêche la parallélisation complète de l'entraînement.
- **Oubli à long terme** : Malgré les portes du LSTM, l'information du début de phrase peut se perdre sur de très longues descriptions.
- **Vocabulaire fixe** : Les mots hors vocabulaire (OOV) sont remplacés par <UNK>, perdant de l'information précise.

5.3 Vers les Transformers et l'État de l'Art

Depuis 2017, l'architecture **Transformer** et son mécanisme de *Self-Attention* ont révolutionné le domaine. Contrairement aux RNN, les Transformers traitent toute la séquence en parallèle et capturent des dépendances à très longue portée. Pour l'image captioning, l'état de l'art actuel repose souvent sur :

- **Vision Transformers (ViT)** : Remplacent le CNN par un Transformer visuel qui découpe l'image en "patches".
- **Modèles Multimodaux (ex: CLIP, BLIP)** : Ces modèles sont pré-entraînés sur des datasets gigantesques (400M+ paires image-texte) pour apprendre un espace latent commun entre vision et langage, permettant des performances "zero-shot" impressionnantes.

6 Conclusion

Ce TP nous a permis de maîtriser les concepts fondamentaux du Deep Learning multimodal. Nous avons réussi à implémenter un pipeline complet : chargement de données, transfert learning avec ResNet, module d'attention personnalisé et génération de texte avec LSTM. Les résultats obtenus confirment l'efficacité de l'attention pour "guider" la génération de texte en se basant sur les caractéristiques visuelles. Cette architecture constitue la base historique sur laquelle se sont construits les modèles génératifs modernes.