Implementation CLIP

Corentin Clément, Victor Buthod, Guillaume Charvolin

19 fevrier 2024

Majeure SCIA-G

EPITA

Table des matières

1 Introduction

Le modèle CLIP (*Contrastive Language–Image Pretraining*) proposé par OpenAI a révolutionné la manière d'apprendre des représentations visuelles et textuelles partagées. L'idée principale est de créer deux espaces latents alignés : un pour les images et un pour les textes, permettant ainsi des tâches de correspondance telles que la récupération d'images à partir de descriptions textuelles ou la description automatique d'images.
Dans ce projet, nous reproduisons et adaptons l'approche décrite dans le papier original. Trois variantes d'architectures ont été testées :

- **BERT + ResNet :** Combinaison de BERT pour le texte et de ResNet pour les images.
- **BERT + ViT :** Utilisation de BERT pour le texte et de ViT (Vision Transformer) pour les images.
- **GPT + ViT :** GPT-2 comme encodeur de texte et ViT pour les images.

Chaque modèle est entraîné en utilisant une fonction de perte contrastive (*contrastive loss*) pour maximiser la similarité entre les paires image-texte associées et minimiser celle entre des paires non correspondantes.

2 Dataset et Prétraitement

2.1 Présentation du dataset

Le dataset utilisé, **Fashion Product Images Small**, contient environ 44,000 images de produits de mode, accompagnées de descriptions textuelles issues de la colonne `productDisplayName`. Ces descriptions sont généralement simples, comme "T-shirt rouge pour homme". Bien que ce dataset soit utile pour des tâches basiques, il présente des limitations :

- Descriptions textuelles redondantes ou vagues, ne permettant pas toujours une distinction claire entre les images.
- Absence d'informations complexes ou contextuelles, nécessaires pour capturer la richesse des modèles GPT.

2.2 Prétraitement des données

Les étapes de prétraitement suivantes ont été appliquées :

- Les images ont été redimensionnées à 224×224 pour être compatibles avec les architectures ResNet et ViT.
- Les pixels ont été normalisés entre -1 et 1.
- Les descriptions textuelles ont été directement utilisées sans enrichissement.

Contrairement au CLIP original, aucune étape de séparation entre jeu d'entraînement et jeu de test n'a été réalisée en raison de contraintes de temps et de ressources.

3 Architecture des Modèles

Chaque modèle CLIP comprend deux encodeurs indépendants : un pour les images et un pour les textes. Ces encodeurs projettent les entrées dans un espace latent partagé de dimension 128, en passant par une normalisation L2. Une température (*learnable parameter*) est utilisée pour ajuster la distribution des similarités pendant l'entraînement.

3.1 Encodeurs d'image

Deux architectures ont été testées pour encoder les images :

- **ResNet :** Un modèle basé sur des convolutions, pré-entraîné sur ImageNet.
- **ViT :** Vision Transformer, qui divise une image en patchs et les traite comme une séquence similaire à un texte.

3.2 Encodeurs de texte

Pour encoder les descriptions textuelles, nous avons testé deux modèles :

- **BERT :** Un modèle bidirectionnel pré-entraîné sur des tâches de compréhension du langage.
- **GPT-2 :** Un modèle génératif unidirectionnel, capable de produire des séquences de texte.

4 Expériences et Résultats

4.1 Entraînement

Les trois modèles ont été entraînés pendant 5 époques sur l'intégralité du dataset en utilisant une taille de batch de 64. L'entraînement a été réalisé sur Google Colab, ce qui a limité le nombre d'époques et la possibilité d'explorer davantage d'hyperparamètres.

4.2 Évaluation

Les performances ont été évaluées selon deux métriques :

- **Accuracy :** Pourcentage de paires image-texte où le texte correct est celui avec la similarité maximale.
- **Top-100 Accuracy :** Pourcentage de paires où le texte correct figure parmi les 100 textes les plus similaires.

Modèle	Accuracy (%)	Top-100 Accuracy (%)
BERT + ResNet	0.74	34.51
BERT + ViT	0.87	37.11
GPT + ViT	0.00	0.10

TABLE 1 – Comparaison des performances des trois modèles.

4.3 Analyse des résultats

Les modèles basés sur BERT ont obtenu des résultats supérieurs, en particulier lorsqu'il est combiné avec ViT. Cela peut s'expliquer par :

- **BERT :** Pré-entraîné sur des tâches de compréhension du langage, il est mieux adapté à des descriptions textuelles simples.
- **ViT :** Plus performant que ResNet pour capturer les caractéristiques visuelles grâce à son approche par patchs.

En revanche, GPT + ViT a échoué à produire des résultats significatifs. Cela s'explique par :

- **Simplicité des descriptions textuelles :** GPT, conçu pour manipuler des séquences complexes, n'a pas pu tirer parti de descriptions simples.
- **Alignement difficile :** GPT étant génératif, son alignement avec les représentations visuelles est plus complexe que celui de BERT.
- **Sous-entraînement :** Cinq époques sont insuffisantes pour entraîner un modèle aussi complexe.

5 Simplifications et Limites

Plusieurs simplifications ont été faites pour respecter les contraintes :

- **Descriptions textuelles simples :** Utilisation directe de `productDisplayName`.
- **Absence de séparation train/test :** Le modèle a été évalué sur le même ensemble utilisé pour l'entraînement.
- **Nombre d'époques limité :** Le faible nombre d'époques a impacté la capacité du modèle à converger.

Ces simplifications expliquent en partie les performances modestes, en particulier pour le modèle GPT + ViT.

6 Conclusion

Ce projet a permis de reproduire une version simplifiée de CLIP et de tester trois architectures. Les résultats montrent que BERT + ViT est la meilleure combinaison dans ce contexte, avec un équilibre entre efficacité et précision. Les performances pourraient être améliorées avec :

- Un dataset enrichi avec des descriptions textuelles plus riches.
- Une augmentation du temps d'entraînement et une exploration des hyperparamètres.
- Une séparation explicite entre les jeux d'entraînement et de test.

Ce travail met en évidence les forces et les faiblesses de l'approche CLIP dans des contextes limités, tout en ouvrant la voie à des améliorations futures.