Bouchard Benjamin

OpenClassroom

Juillet 2022

INTRODUCTION: Une présentation du modèle Vision Transformer

Les modèles de type Transformer, basés sur le mécanisme d'Attention, sont récemment devenus incontournables dans le domaine du Natural Language Processing (NLP) et sont devenus les standards de facto. En revanche, dans le domaine de la "Computer Vision", les réseaux de neurones convolutifs (CNNs) restent prédominants. Toutefois, récemment plusieurs tentatives d'utiliser le mécanisme d'Attention pour la vision par ordinateur ont été menées. L'une d'entre elles se montre prometteuse et a abouti au modèle VIson Transformer (ViT) que nous nous proposons ici de présenter succinctement.

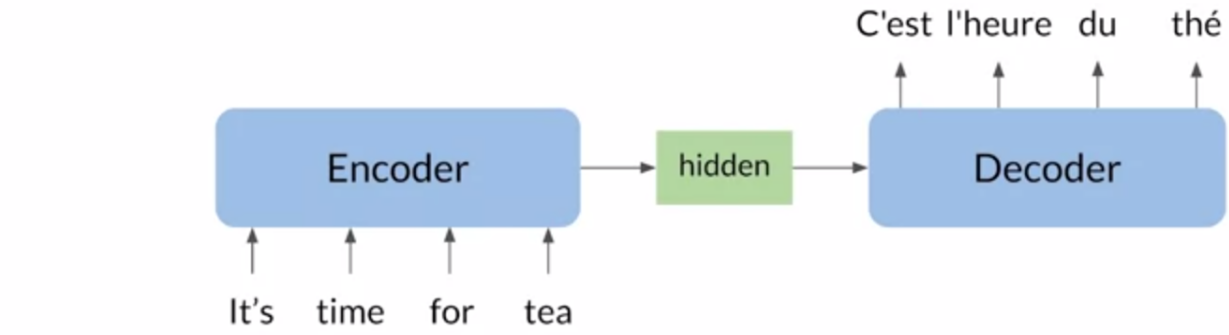
Les modèles de transduction de séquences classiques et le mécanisme d'Attention.

Le mécanisme d'Attention a été introduit en NLP dans les modèles dits de "transduction de séquences", c'est à dire les modèles prenant en entrée une séquence et donnant une séquence en sortie (typiquement les modèles permettant la traduction d'une langue dans une autre pour lesquels on dispose en entrée d'une séquence de tokens dans une langue et qui donnent une séquence de tokens dans une autre langue en sortie).

Ces modèles sont constitués de deux parties:

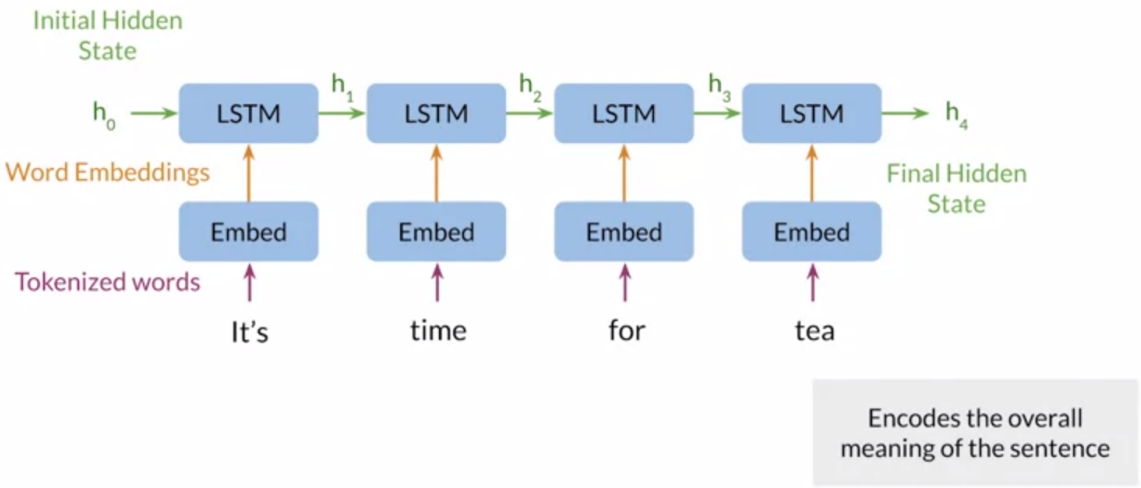
- un encodeur

- un décodeur



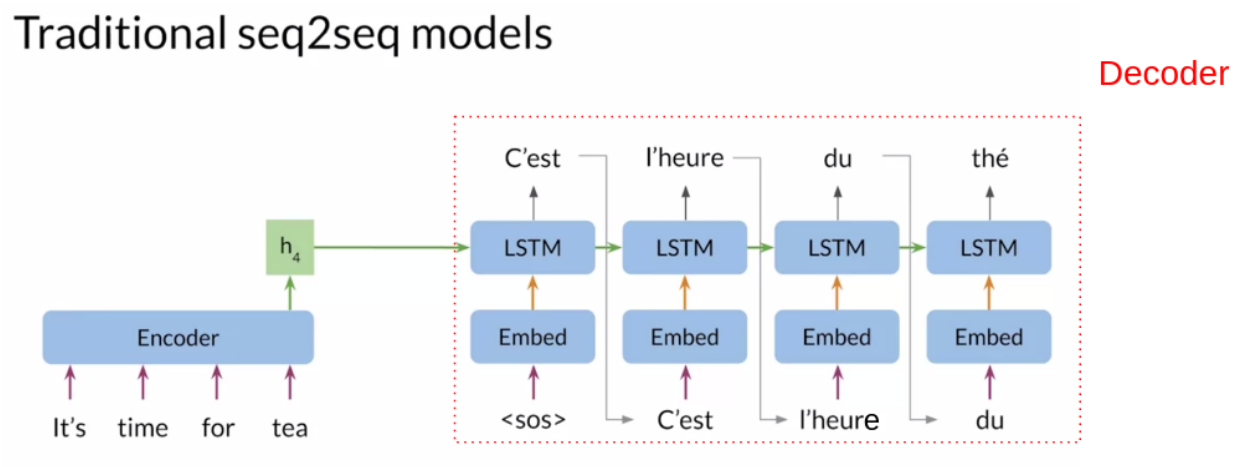
© Copie d'écran de la vidéo du cours attention-models-in-nlp de Coursera.

Ces deux parties ont des structures similaires. Ci-dessous est représentée la structure d'un encodeur:



© Copie d'écran de la vidéo du cours attention-models-in-nlp de Coursera.

Ces deux blocs (encodeur/décodeur) sont combinés de la façon suivante:

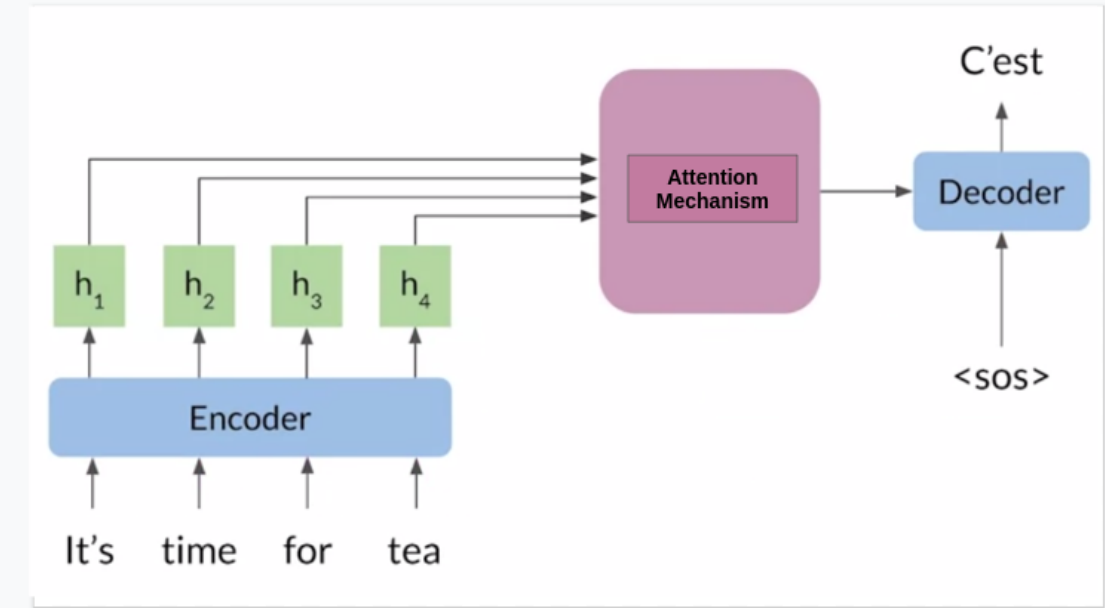


© Copie d'écran de la vidéo du cours attention-models-in-nlp de Coursera.

LSTM est l'acronyme de "Long Short Term Memory". Il s'agit d'un type de réseau de neurones récurrent qui permet de maintenir des dépendances entre les blocs d'entrée, qui seront utilisées lors de l'élaboration des séquences de sortie.

Si ce genre de modèle fonctionne bien sur des séquences relativement courtes, ce n'est plus vrai si la longueur de la séquence d'entrée s'allonge. Ces modèles ne permettent pas de maintenir des dépendances pertinentes entre des tokens qui sont éloignés dans la séquence d'entrée. Et ce parce que toute l'information sur les dépendances dans la séquence d'entrée est communiquée au décodeur uniquement par le dernier état caché de l'encodeur (h4 dans le schéma ci-dessus)

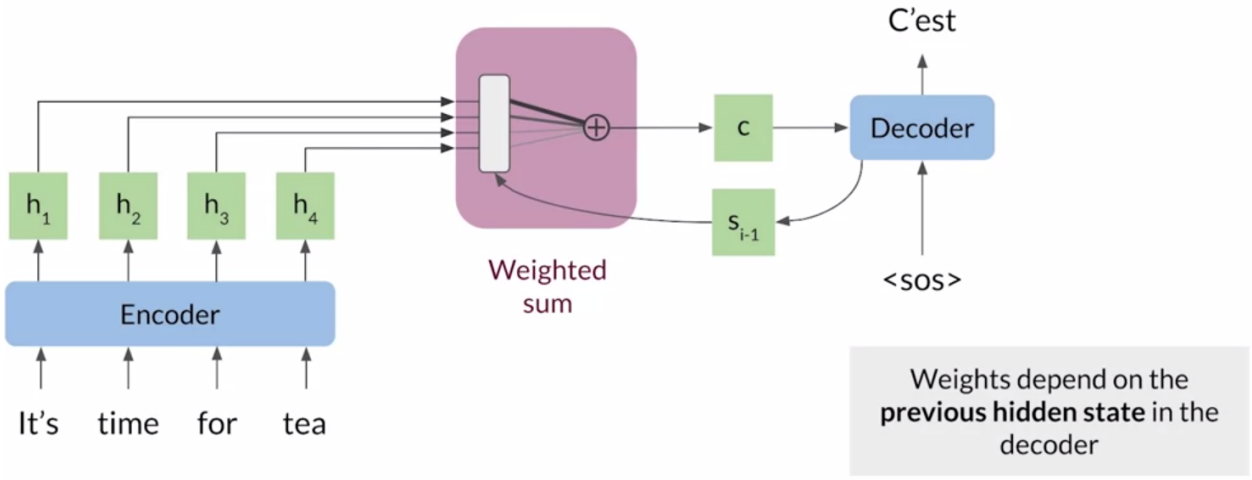
Le mécanisme d'Attention vise à corriger ce défaut:



© Copie d'écran de la vidéo du cours attention-models-in-nlp de Coursera.

En ajoutant le bloc d'Attention, on parvient à focaliser l'attention du décodeur sur les tokens d'entrée pertinents au fur et à mesure de l'élaboration de la séquence de sortie.

Le bloc d'Attention peut être décrit comme un réseau de type "Feed-Forward" dont les poids sont dynamiques (et qui n'évoluent pas uniquement de façon graduelle à chaque "epoch" selon le mécanisme de back-propagation comme lors de la phase d'entraînement d'un réseau classique).



© Copie d'écran de la vidéo du cours attention-models-in-nlp de Coursera.

All you need is Attention.

En 2017, un tout nouveau modèle entièrement basé sur le mécanisme d'Attention a été introduit.

Ce nouveau modèle, appelé "Transformer", est décrit par le schéma suivant:

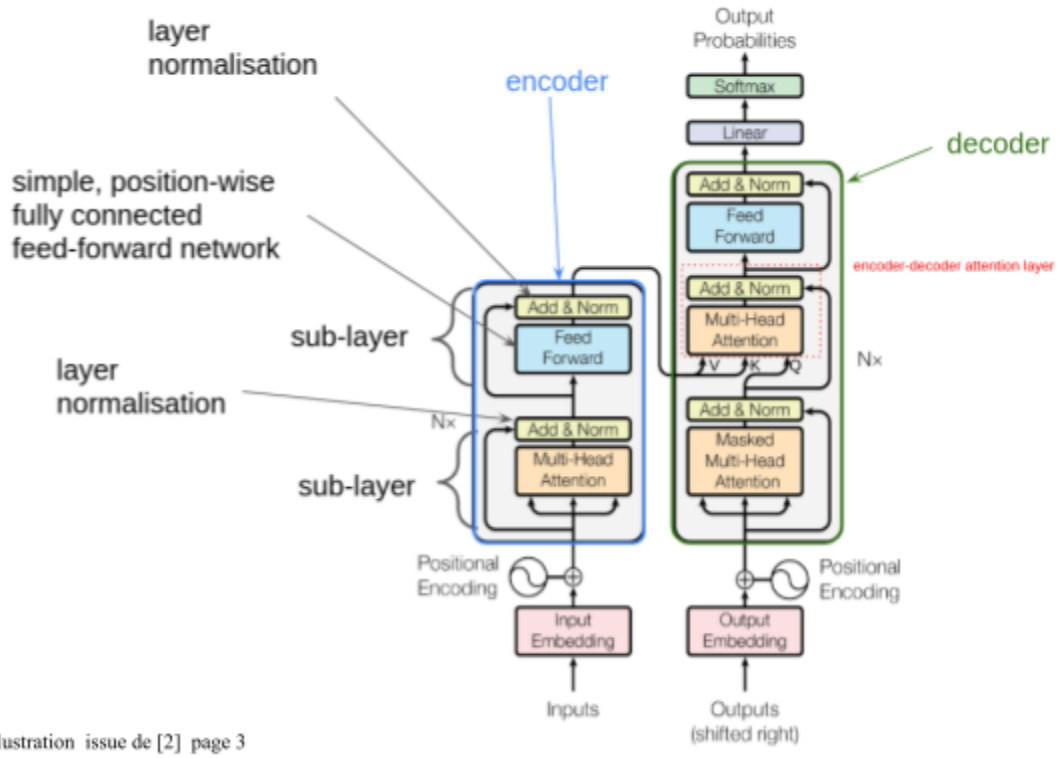
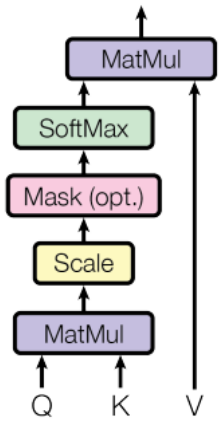


Illustration issue de [2] page 3

Dans ce modèle, “Inputs” correspond à la séquence d’entrée dans son ensemble et la partie “Outputs” correspond la séquence de sortie en cours d'élaboration (incomplète). “Inputs” et “Outputs” sont encodées (embedding plus des informations de position). Ils sont ensuite chacun passés par un bloc d’Attention. Les sorties de ces deux blocs d’Attention sont ensuite réunis dans un troisième bloc d’Attention dans la couche “encoder-decoder attention”. La partie “encoder” fournissant les composants K et V (Key et Value) et la partie “decoder” (basse) fournissant la composante Q (Query)

Le calcul qui a lieu dans un bloc d’Attention est décrit par:

Scaled Dot-Product Attention



ce qui est équivalent à:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Illustrations issue de [2] page 4

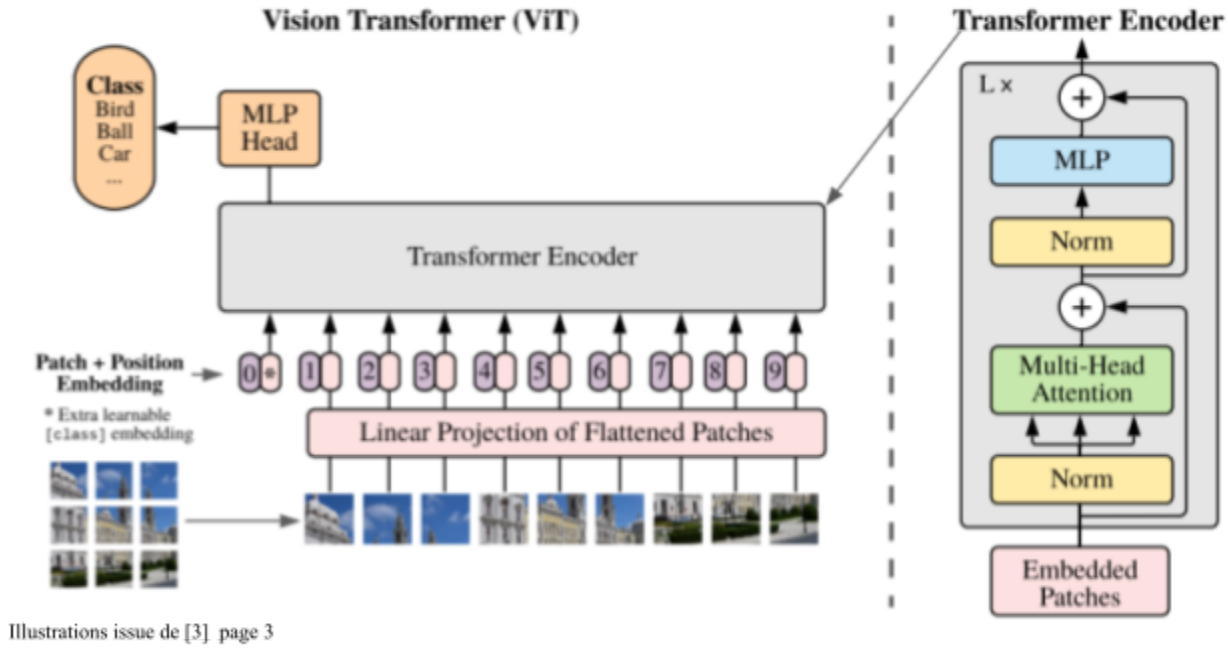
Intuitivement, ce mécanisme se comprend de la sorte: on cherche à produire un nouveau token pour la séquence de sortie, ce qui correspond à la query Q. Pour produire ce nouveau token, on utilise l’information produite par l’encodeur c’est-à-dire les paires clé-valeur, l’accès aux valeurs

V à partir de la requête Q se faisant par l'intermédiaire des clés K. Ce nouveau mécanisme d'Attention est appelé "self-attention"

Le mécanisme d'Attention ainsi décrit correspond en fait à une seule tête d'Attention. En pratique, ce qui est utilisé c'est un mécanisme d'Attention "multi-head".

Le modèle Vision Transformer.

Il a été proposé de s'inspirer de ce qui a été fait dans le domaine de la NLP et de l'adapter au domaine de la vision par ordinateur. Voici une description du modèle qui a été proposé:



Illustrations issue de [3] page 3

Dans ce modèle, on retrouve le bloc "encoder" sur la droite utilisé dans le modèle de NLP, le bloc MLP (MultiLayer Perceptron) étant l'équivalent du bloc "Feed Forward".

Ce schéma se comprend de la sorte:

- L'image à traiter est découpée en blocs de 16x16 pixels appelés patches,
- Ces patchs sont ensuite disposés dans l'ordre (de la gauche vers la droite puis de haut en bas) et sont aplatis, c'est-à-dire que les tensors (plusieurs dimensions) sont transformés en vecteur (une seule dimension),
- Ces vecteurs sont ensuite projetés, et ces projections sont combinées avec des informations de position auquel on ajoute également l'étiquette associée à l'image,
- Ces données sont alors soumises en entrée du bloc "Transformer-Encoder",
- En sortie, un simple réseau de type "softmax" permet d'effectuer la classification.

Plusieurs ont été variants de cette architecture ont été proposés:

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

Tableau issue de [3] page 5

Les modèles de type ViT se sont montrés plus performants que l'état de l'art dans tous les cas étudiés:

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k

Etat de l'art

Tableau issue de [3] page 6

Ils nécessitent également moins de temps pour être entraînés que l'état de l'art. Mais il s'agit de ressources nécessaires à l'étape de "fine-tuning", c'est-à-dire l'adaptation d'un modèle déjà entraîné à un jeu de données spécifique. En revanche, pour l'entraînement préliminaire les modèles de type ViT nécessitent plus de données que l'état de l'art pour parvenir à de tels résultats.

Mise en place d'un benchmark:

Modèles comparés:

Nous avons décidé de comparer le modèle ViT avec un modèle de type ResNet (réseau de neurones résiduel). Notre choix s'est porté sur le réseau ResNet50 présent dans la librairie Keras: Resnet Keras. Le modèle ViT disponible sur le Git officiel (vision_transformer) est développé en Flax/JAX. Mais un modèle compatible TensorFlow est disponible ici: ViT-fe-s16 (page complète: ViT-jax2tf).

Nous avons effectué du Transfert Learning et donc utilisé des modèles pré-entraînés sur ImageNet. ImageNet est un dataset qui comprend 14 millions d'images. Nous n'avons donc utilisé que la partie "feature extraction" de ces deux modèles auxquelles nous avons adjoint un réseau dense en sortie. A noter que dans le cas du modèle ViT, ce réseau de sortie se résume à un simple "SoftMax".

D'après, les informations disponibles sur les pages de ces modèles, les résultats sur le **dataset ImageNet complet** sont les suivants:

Modèle	Top-1 accuracy
ResNet50	74.9%
ViT	80.5%

Dataset étudié:

Nous avons repris le dataset du projet 6 d'Openclass Room: Standford Dogs Data Set

Il s'agit d'un jeu de données de 20 580 images comprenant 120 races de chien issues d'ImageNet.

Objet du benchmark:

Outre de comparer les deux modèles sur le dataset cible, notre benchmark va permettre de comparer les performances des deux modèles dans le cadre de "fine-tuning".

Résultats obtenus:

Sur le dataset de notre étude, les résultats obtenus sont les suivants:

Modèle	Accuracy (test)	Temps (User time) GPU
ResNet 50	70.8%	5h 6min 47s
ViT	80.8%	4h 43min 50s

Analyse des résultats:

Le modèle ViT présente un avantage de 10 points sur le modèle ResNet 50 et a pris moins de temps pour être fine-tuné (-7%). Par ailleurs, sa performance sur une tâche spécifique (dataset de chiens) se trouve proche et même meilleure que celle observée sur le dataset complet (80.8% versus 80.5%).

CONCLUSION

Les modèles de type ViT qui s'inspirent du mécanisme d'Attention issu du domaine de la NLP s'annoncent prometteurs et semblent pouvoir concurrencer les modèles de type CNN (classiques ou résiduels) qui constituent jusqu'à maintenant l'état de l'art dans le domaine de la computer vision. En particulier, ces modèles semblent être particulièrement indiqués pour le "fine-tuning". En revanche, ils réclament beaucoup plus de données dans la phase d'entraînement. Mais comme ils ne montrent pour l'instant pas de signe de saturation et que les techniques de data augmentation semble très bien fonctionner avec eux, cela pourrait ne pas empêcher qu'ils deviennent le nouvel état de l'art dans le domaine de la vision par ordinateur.

Bibliographie

[1] Neural Machine Translation by Jointly Learning to Align and Translate

[2] Attention Is All You Need

[3] An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale