CENTRALE LYON

Large Language Models & ChatGPT

MOD 2.1: Défis Informatiques du Big Data

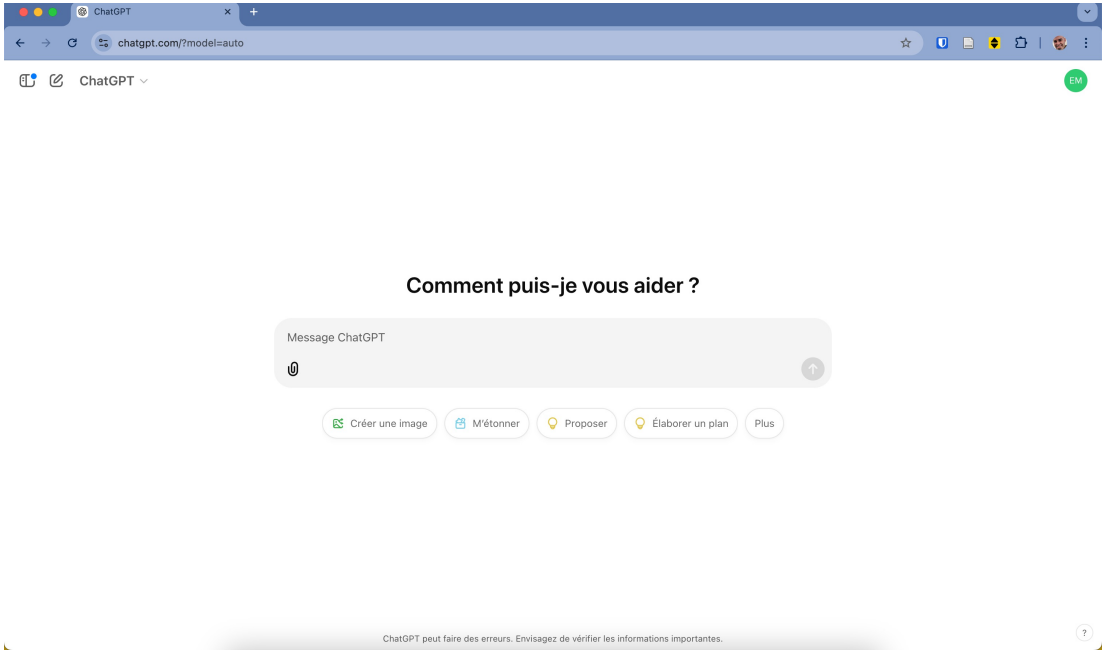
Emmanuel Dellandréa

emmanuel.dellandrea@ec-lyon.fr

2024-2025

LLM & ChatGPT

- Introduction aux modèles de langage (LLM)
- Les Transformers
- Principe de ChatGPT & autres LLM
- Exemples d’utilisation de ChatGPT



ChatGPT ?

- Un agent conversationnel basé sur un modèle de langage développé par la société OpenAI
- Système génératif de texte s'appuyant sur un prompt (texte d'entrée)
- Son objectif : prédire à chaque étape le mot suivant d'une séquence de mots
- Comment y avoir accès ? ➔ https://chatgpt.com/
  ou applications mobiles

➔ Quelques exemples d'utilisation

- Distance entre la Terre et la Lune ?
- Résumé de la vie de Charlemagne
- Recette de cuisine
- Ecriture d'un email
- Réécriture d'un texte
- Ecriture d'un discours pour une entretien d'embauche

Modèle de langage

- Objectif : modéliser les structure syntaxiques et grammaticales d'une langue

- Applications nombreuses :
  - Reconnaissance de la parole
  - Traduction automatique
  - Compréhension de texte
  - Génération de texte (par exemple Q/A ou résumés)
  - Détection de paraphrases
  - Analyse de sentiments

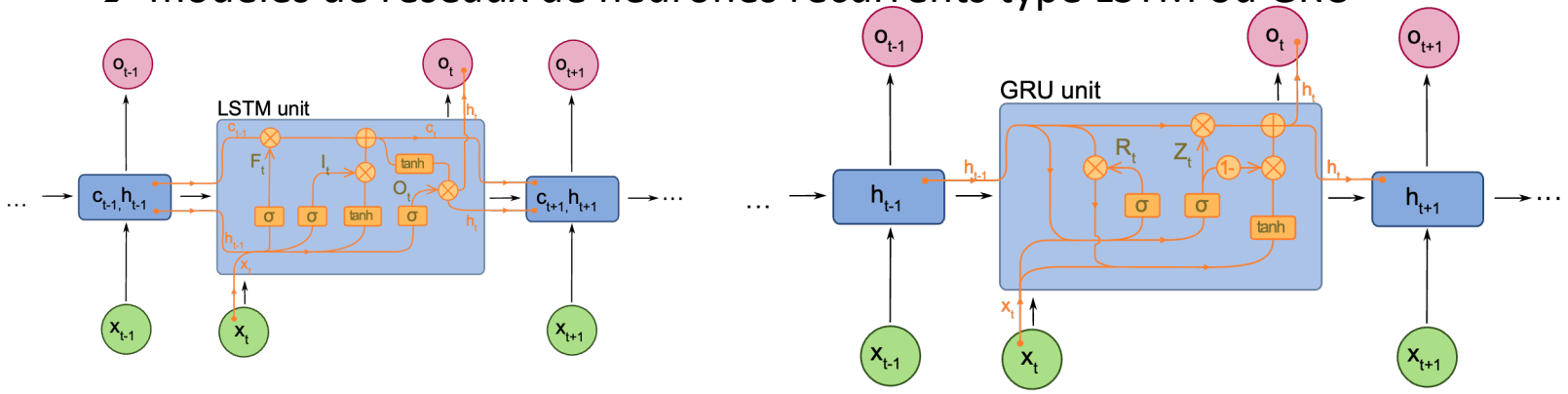
  …

Modèle de langage

CENTRALE LYON

Principaux modèles (jusqu'en 2017) :

➔ modèles de réseaux de neurones récurrents type LSTM ou GRU



Source : Wikipedia https://commons.wikimedia.org/w/index.php?curid=60466441

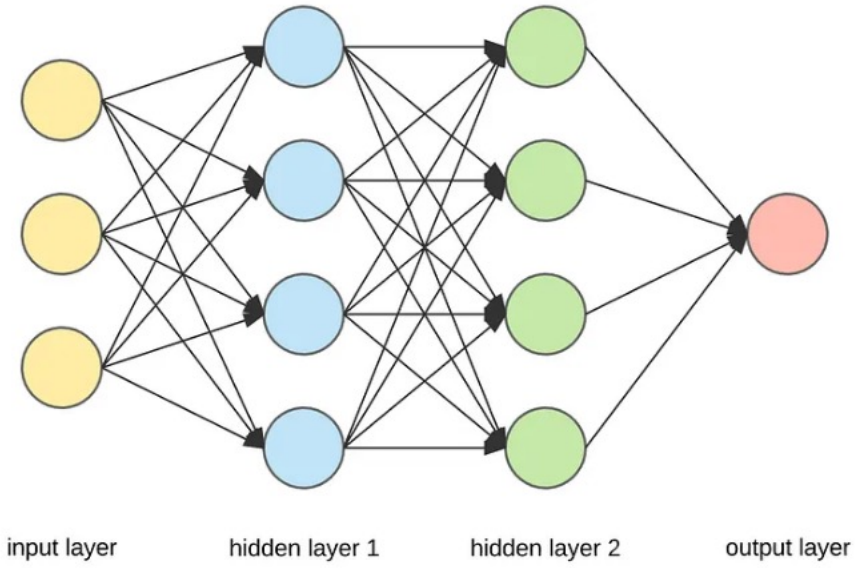
Limites de ces modèles :

- Difficultés à traiter des séquences longues
- Problèmes fréquents de surapprentissage
- Difficultés à modéliser des relations complexes
- Calculs ne pouvant pas être parallélisés (besoin du mot précédent)

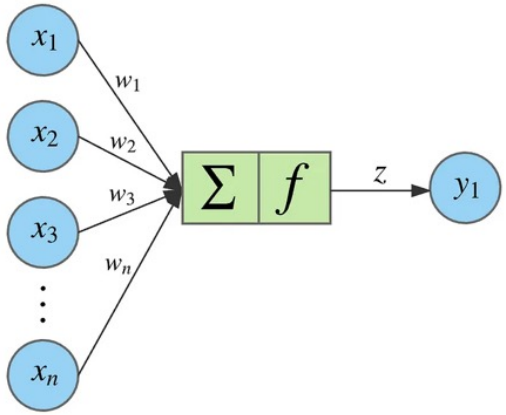
Réseaux de neurones

➔A la base de ces modèles de langage

Réseau de neurones



Modélisation d'un neurone



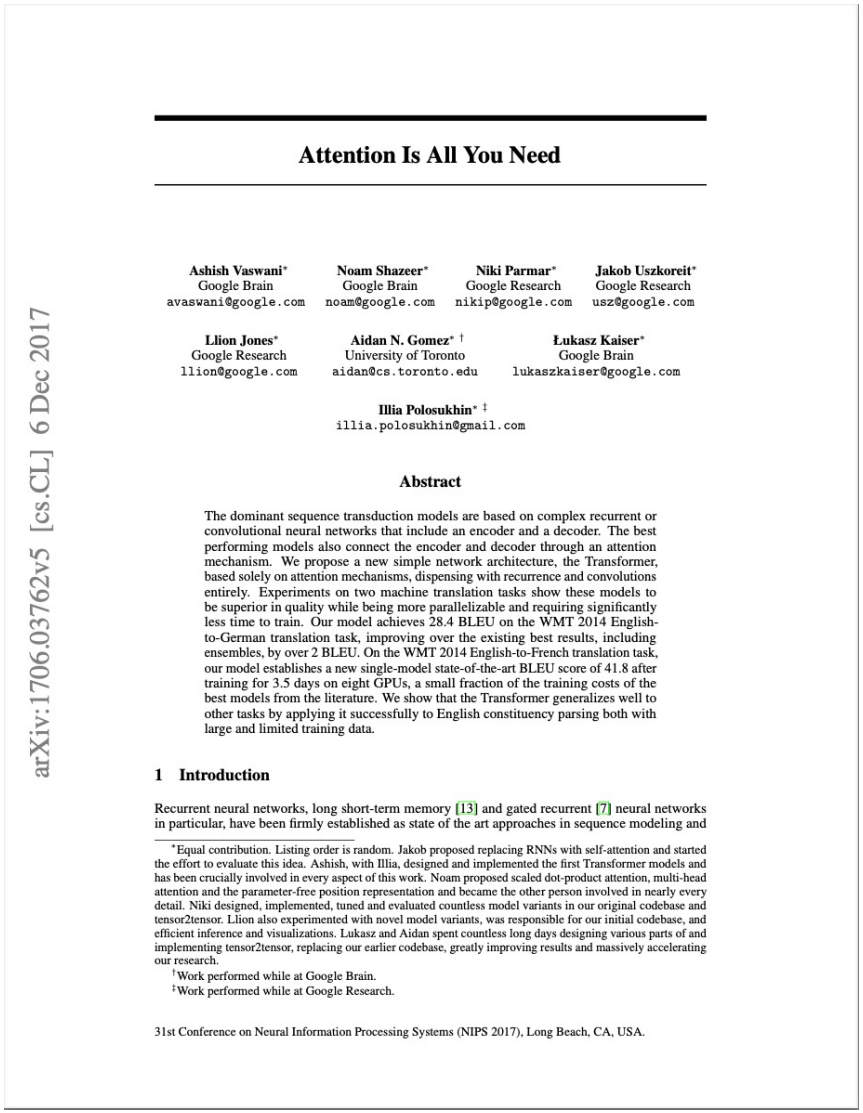
Source : https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6

Transformers

Modèles de langage s'appuyant sur un mécanisme d'auto-attention

Paru en 2017

➔ Révolution dans le domaine du Traitement Automatique du Langage



arXiv:1706.03762v5 [cs.CL] 6 Dec 2017

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

‡Work performed while at Google Research.

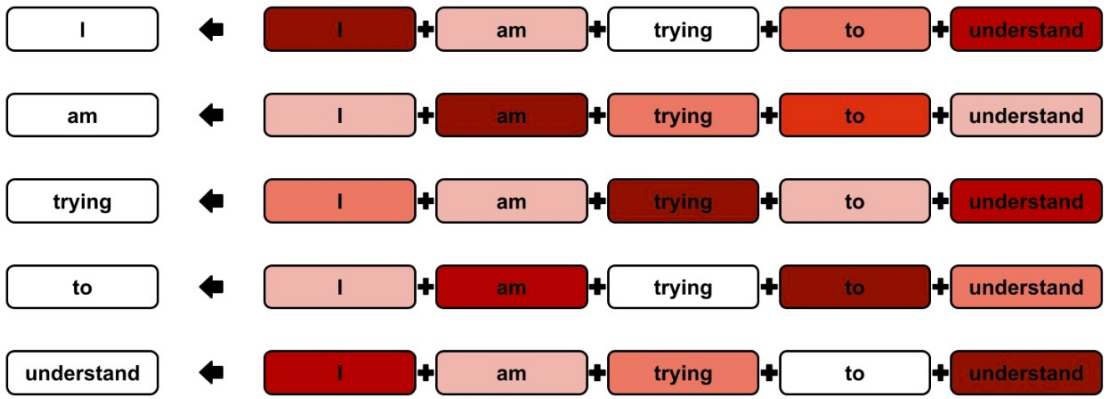
31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Transformers

Principe de l'auto-attention

- Calculer une pondération pour chaque élément de l'entrée, en fonction de sa relation avec tous les autres éléments de l'entrée.
- Utiliser ces pondérations d'attention pour générer des représentations de plus haut niveau pour la séquence, en combinant les informations de chaque élément de la séquence pondérées par leurs poids d'attention.

➔ Permet de capturer des relations complexes entre les différents éléments de la séquence

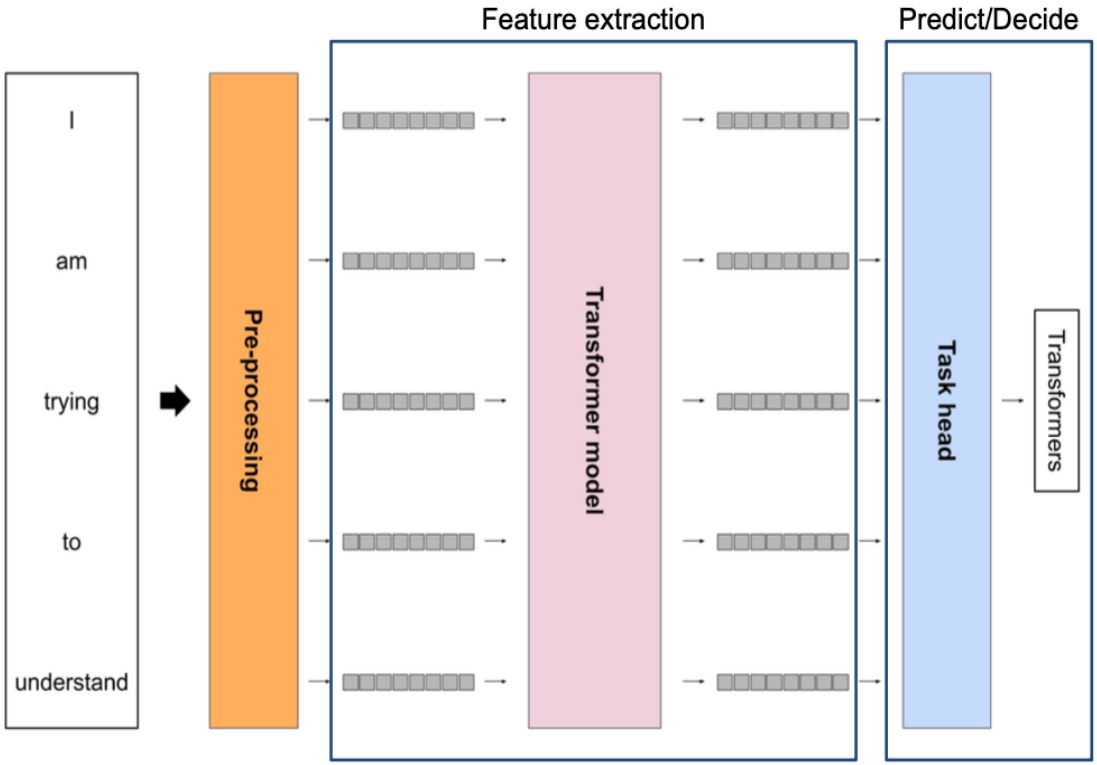


Source : FIDLE, section 8 Transformers, https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle/-/wikis/home

Transformers

CENTRALE LYON

Architecture globale du modèle :

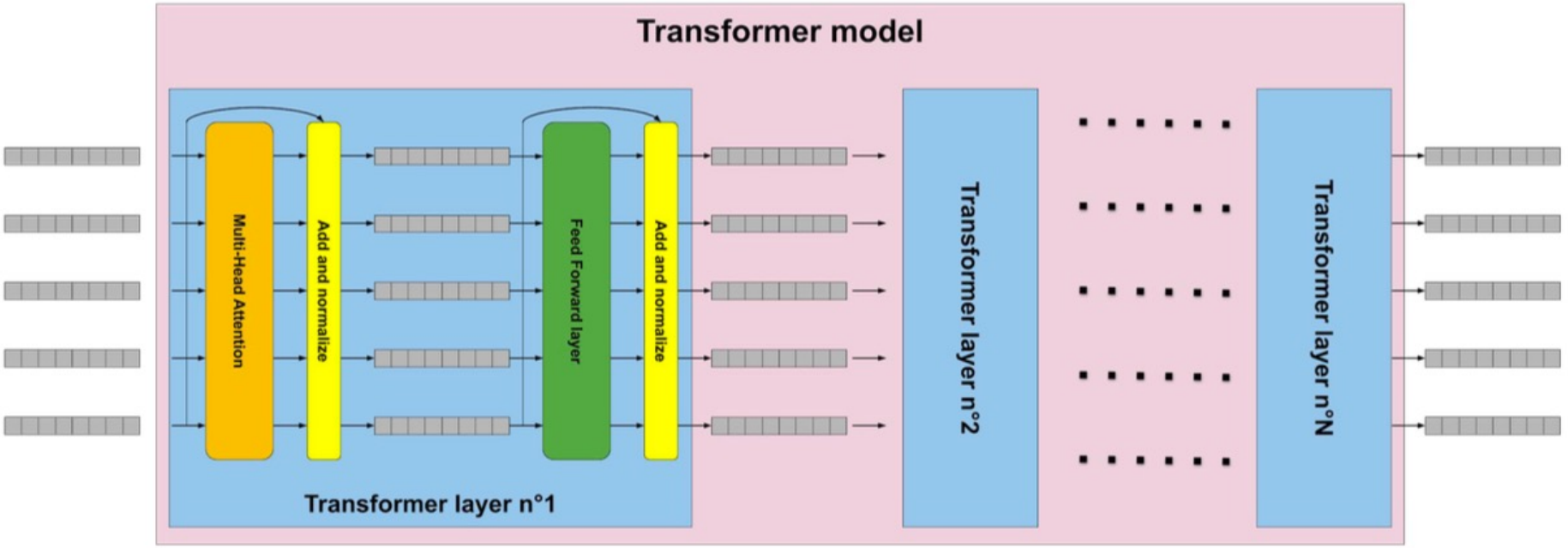


Source : FIDLE, section 8 Transformers, https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle/-/wikis/home

Transformers

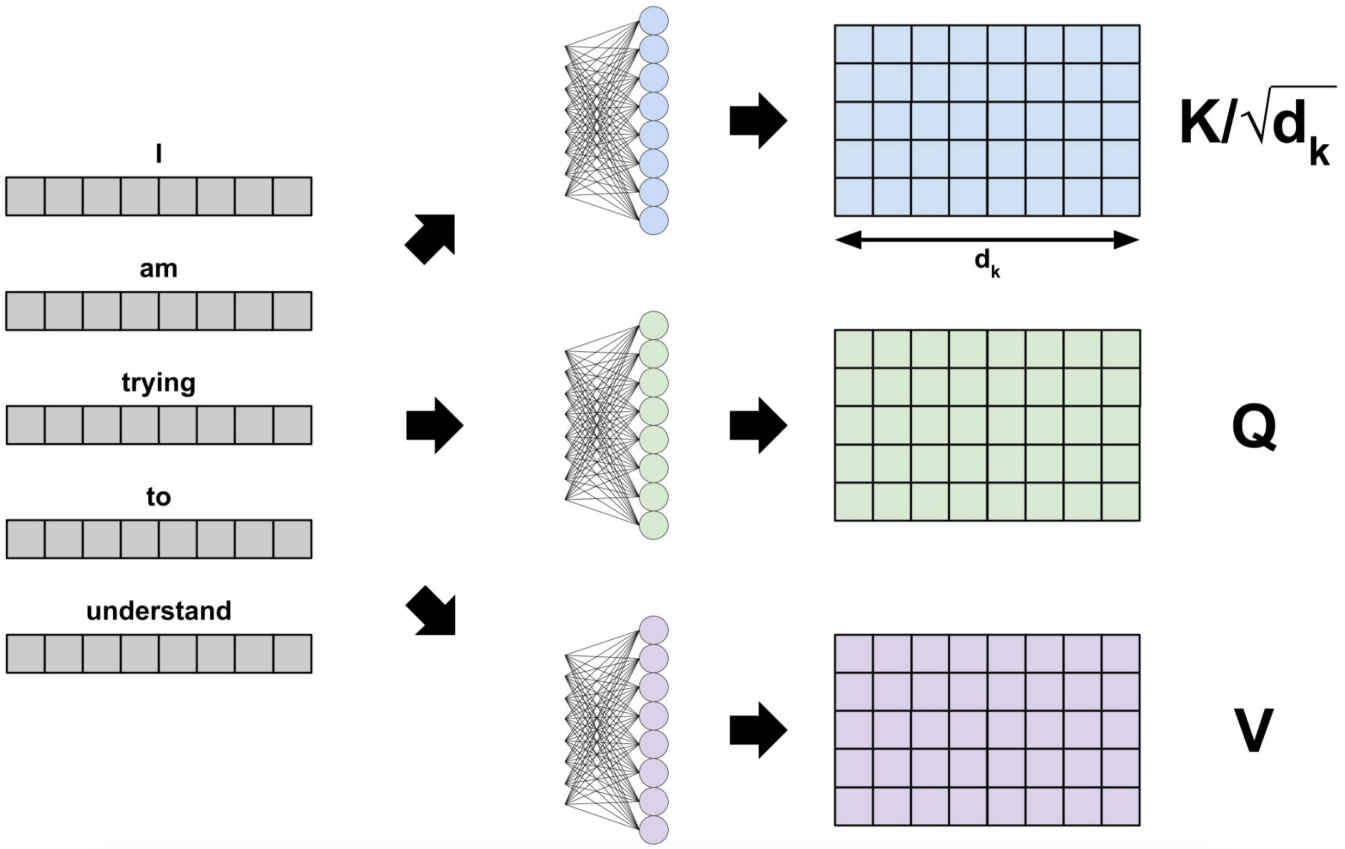
CENTRALE LYON

Architecture du Transformer :



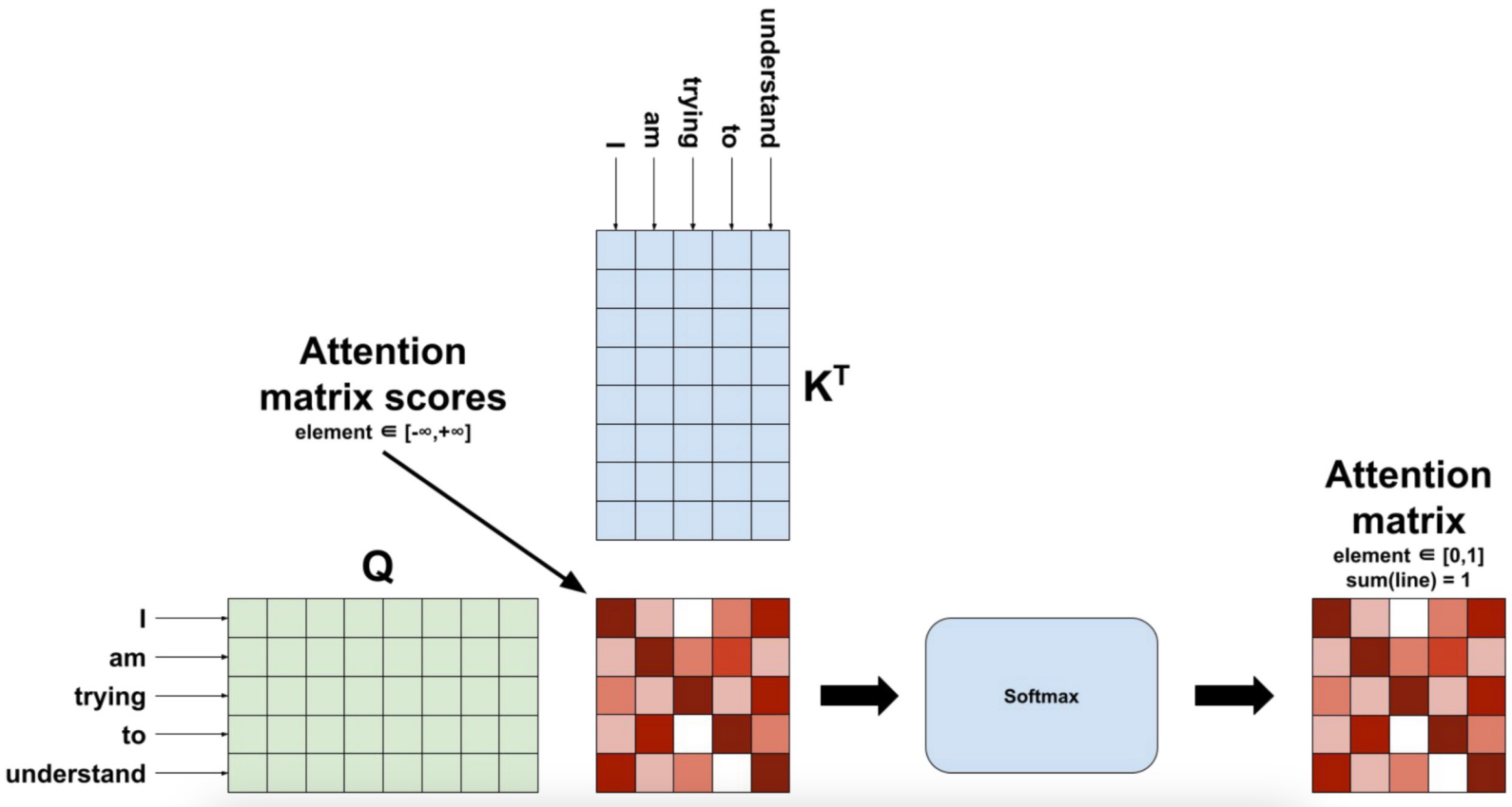
Source : FIDLE, section 8 Transformers, https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle/-/wikis/home

Mécanisme d'auto-attention



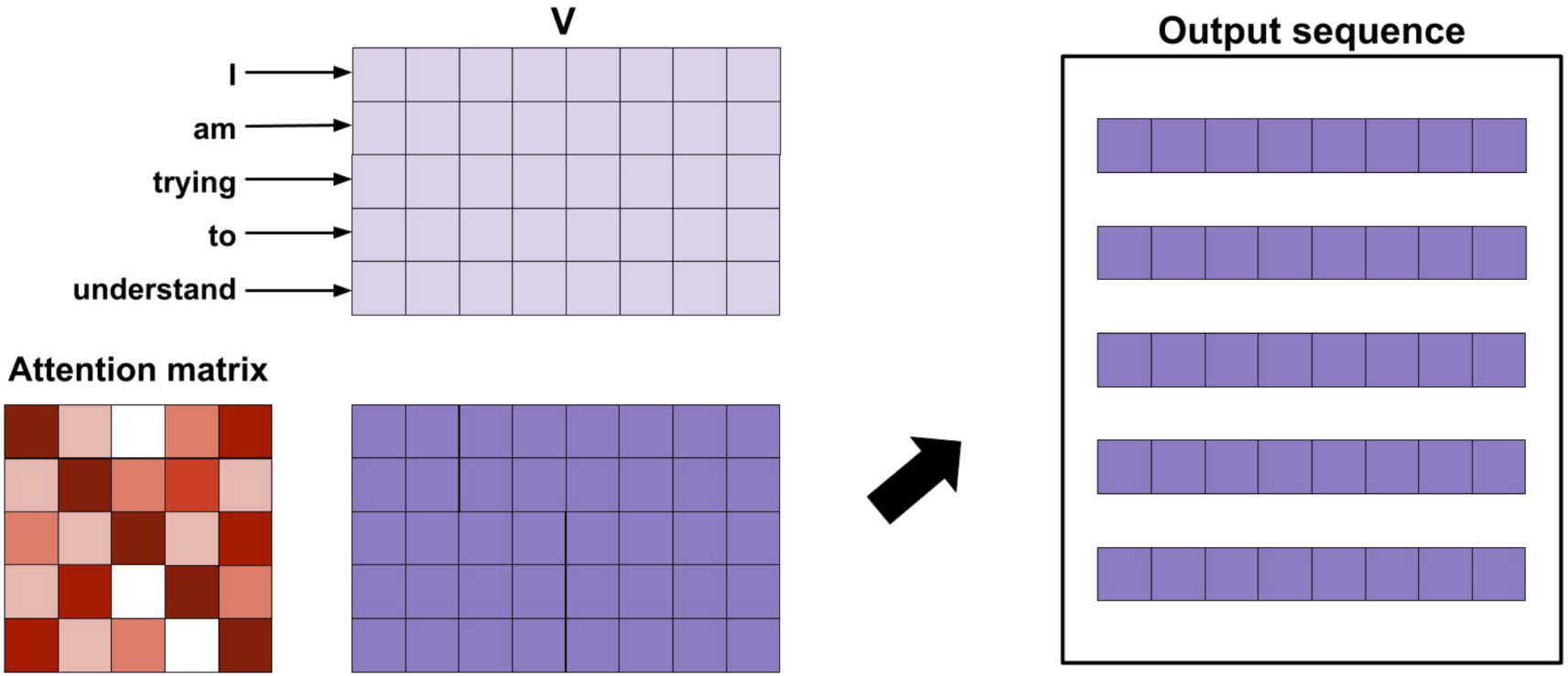
Source : FIDLE, section 8 Transformers, https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle/-/wikis/home

Mécanisme d'auto-attention



Source : FIDLE, section 8 Transformers, https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle/-/wikis/home

Mécanisme d'auto-attention



Source : FIDLE, section 8 Transformers, https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle/-/wikis/home

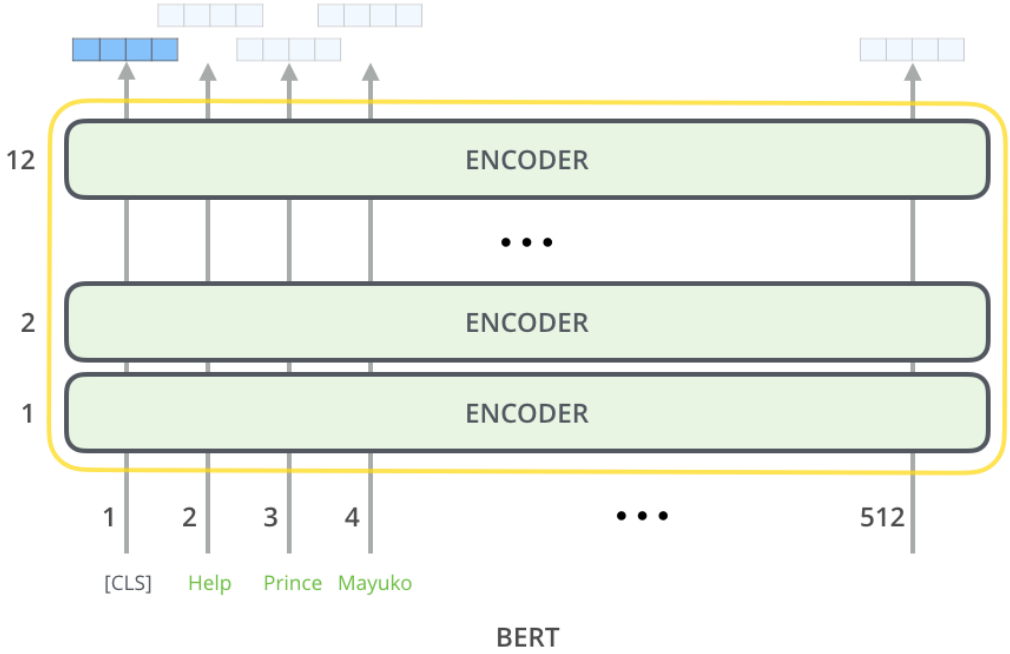
Variantes des Transformers

Encodeur

ex : BERT (Bidirectional Encoder Representations from Transformers)

➔ Pour des applications ne nécessitant pas de génération séquentielle de texte

- Classification de texte
- Reconnaissance de sentiments
- Détection de paraphrases
- …



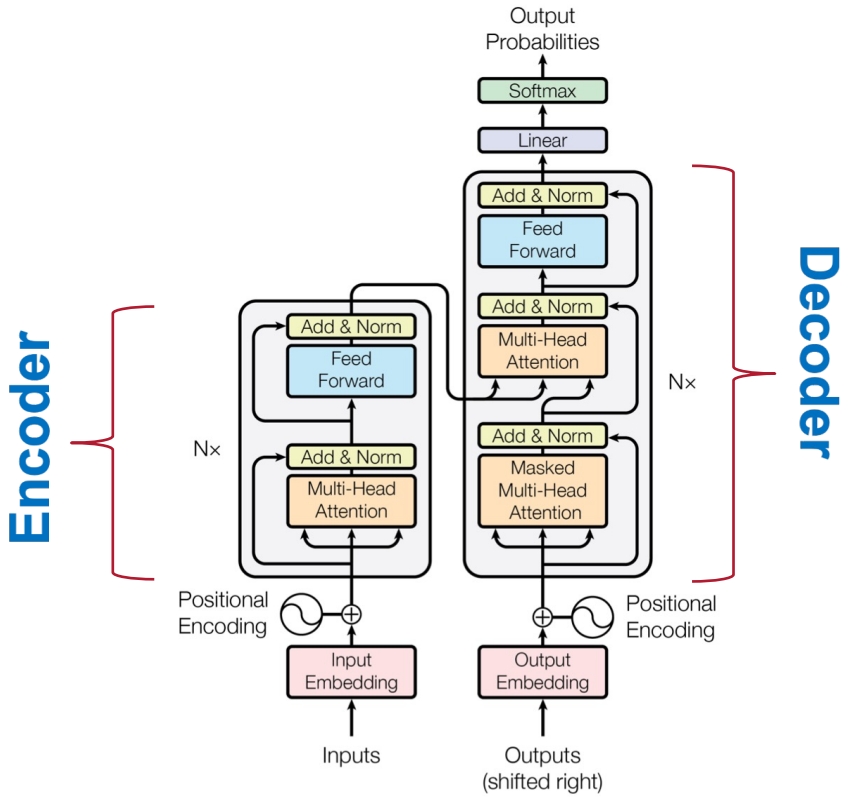
Source : https://jalammar.github.io/illustrated-bert/

Variantes des Transformers

Encodeur-Décodeur

➔ Pour des applications nécessitant la génération séquentielle de texte et la compréhension globale de la séquence d'entrée

- Traduction de texte
- Reformulation de texte
- …



Source : « Attention Is All You Need », Vaswani et al., NIPS 2017

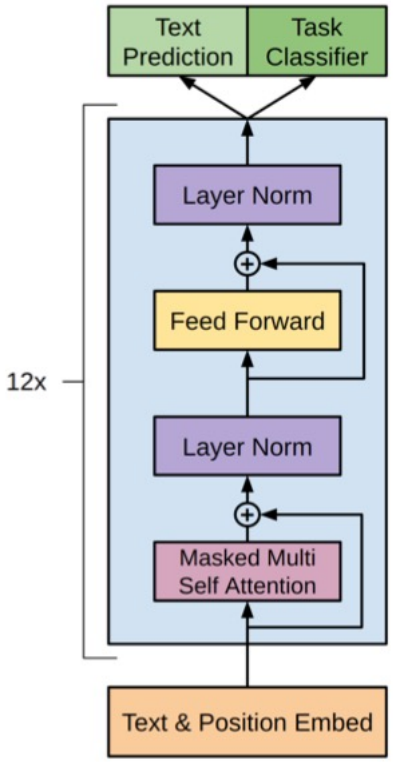
Variantes des Transformers

Décodeur

ex : GPT (Generative Pre-trained Transformer)

➔ Pour des applications nécessitant la génération séquentielle de texte

- Reformulation
- Questions/réponses

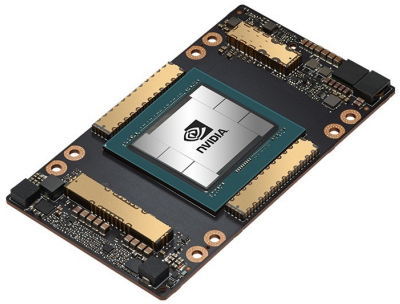


Source : « Improving Language Understanding by Generative Pre-Training », Radford et al., 2018

Accélération matérielle sur GPU

La parallélisation sur GPU des Transformers repose sur leur architecture hautement parallélisable : utilisation intensive de calcul matriciel et de mécanismes d'attention qui peuvent être efficacement distribués sur les nombreuses unités de calcul d'un GPU (technologie CUDA)

- Parallélisation au niveau des opérations matricielles
- Parallélisation sur les dimensions séquentielles et de batch



Transformers & chatGPT

CENTRALE LYON

Genèse de ChatGPT :

2018 : GPT (Improving Language Understanding by Generative Pre-Training)

➔ Utilisation de la partie décodeur d'un transformer pré-entrainé et affiné pour réaliser différentes tâches (117 millions de paramètres)

2019 : GPT-2 (Language Models are Unsupervised Multitask Learners)

➔ Q/A avec un prompt en langage naturel et des réponses proches de celles d'un humain (1,5 milliard de paramètres)

2020 : GPT-3 (Language Models are Few-Shot Learners)

➔ amélioration du modèle précédent avec un réseau beaucoup plus grand (175 milliards de paramètres, 96 couches)

2022 : chatGPT & InstructGPT (Training language models to follow instructions with human feedback)

➔ amélioration du modèle en utilisant un apprentissage supervisé, et par renforcement

2023 : GPT-4 ➔ modèle beaucoup plus grand que le précédent, multimodal (permet l'utilisation d'images et de textes en entrée) (1,8 trillions de paramètres)

ChatGPT

CENTRALE LYON

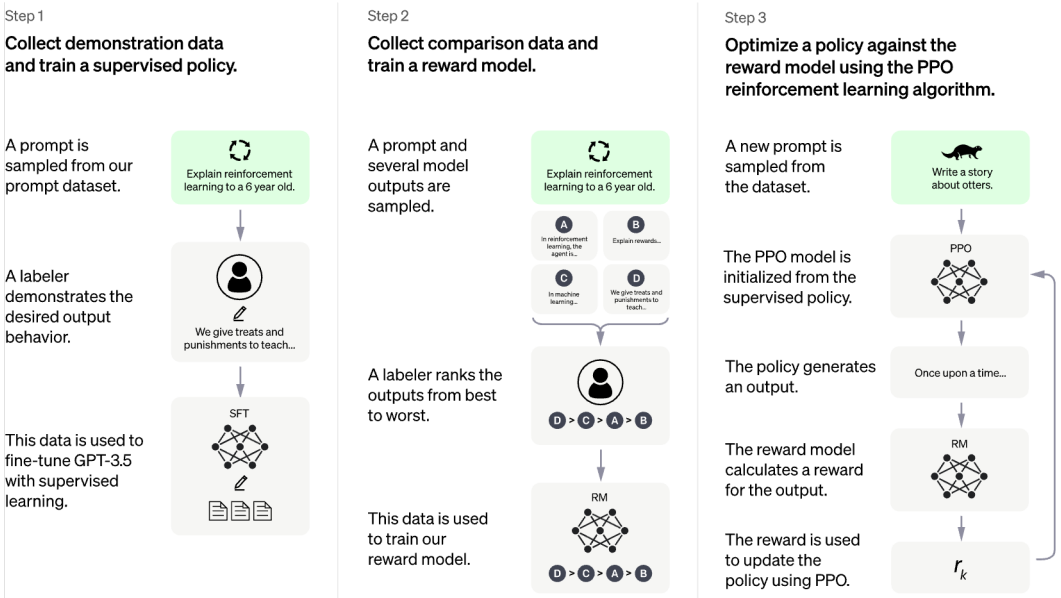
- Caractéristiques (décembre 2024)
  - Basé sur l'architecture GPT-4o mini (GPT-4 et GPT-4o pour la version payante)
  - Données d'apprentissage collectées jusqu'en octobre 2023
  - Disponible en plusieurs langues (anglais, français, espagnol, chinois, …)
  - Supporte des données de type texte, image, vidéo et audio en entrée
  - En sortie : texte, et progressivement image, vidéo, audio
  - 1800 milliards de paramètres dans GPT-4, 200 milliards dans GPT-4o, 8 milliards dans GPT-4o mini

Apprentissage de ChatGPT

CENTRALE LYON

En 4 étapes principales

1) Apprentissage non supervisé du modèle GPT-4 sous-jacent
2) Apprentissage supervisé (Fine-Tuning) du modèle ChatGPT sur des exemples de questions/réponses rédigés par des humains
3) Apprentissage supervisé d'un modèle d'évaluation des réponses
4) Apprentissage par renforcement (Fine-Tuning) utilisant ce modèle d'évaluation pour améliorer les performances de ChatGPT



Source : OpenAI, https://openai.com/blog/chatgpt

Apprentissage de ChatGPT

CENTRALE LYON

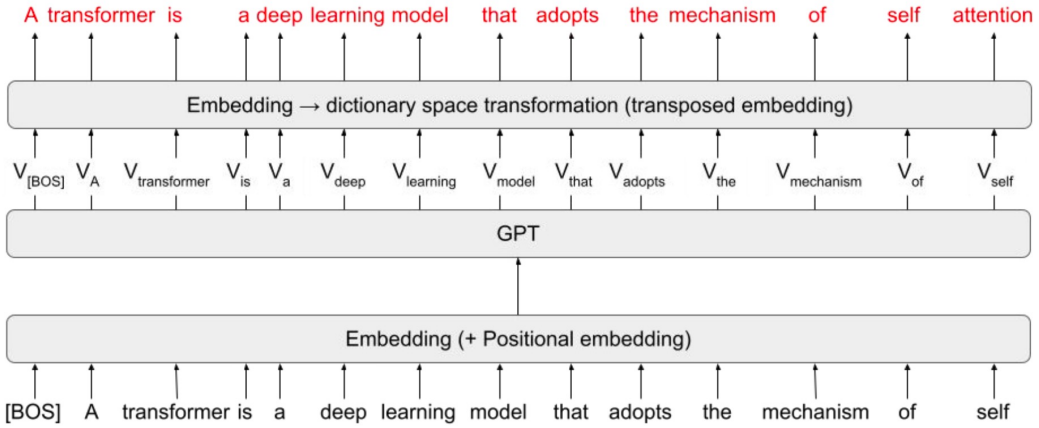
Etape 1 : Apprentissage non supervisé du modèle GPT-4

Objectif : apprendre les structures syntaxiques et grammaticales de la langue, ainsi que des connaissances générales sur de nombreux domaines

Données :

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Stratégie : prédiction étape par étape du mot suivant dans une séquences de mot



Source : FIDLE, section 8 Transformers, https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle/-/wikis/home

Apprentissage de ChatGPT

CENTRALE LYON

Etape 2 : Apprentissage supervisé du modèle ChatGPT (Fine Tuning)

Objectif : adapter le modèle à la tâche de conversation

Données : exemples de séquences de dialogues (questions/réponses) rédigés par des humains

Stratégie : l'apprentissage du modèle pré-entrainé est poursuivi avec ces données

Etape 3 : Apprentissage supervisé d'un modèle d'évaluation des réponses

Objectif : obtenir un modèle qui sera capable d'évaluer les réponses fournies par le modèle génératif, nécessaire pour l'étape 4

Données : exemples de questions avec plusieurs réponses générées par le modèle. Un humain ordonne ces réponses par qualité

Stratégie : un réseau de neurones est appris à partir de ces données

Apprentissage de ChatGPT

CENTRALE LYON

Etape 4 : Apprentissage par renforcement du modèle ChatGPT (Fine Tuning)

Objectif : optimiser le modèle

Données : exemples de questions

Stratégie :

- le modèle ChatGPT génère une réponse pour chaque question choisie
- Le modèle d'évaluation prédit la qualité de cette réponse
- Cette valeur est utilisée comme signal de récompense d'un apprentissage par renforcement (algorithme Proximal Policy Optimization) pour améliorer la qualité de génération du modèle

Quelles limites ?

- Phénomène d'hallucination : ChatGPT peut générer des contre-vérités présentées de manière très cohérente et convaincante (et même de fausses sources)
- Apporte des réponses non sourcées (inadapté à la recherche d'informations), et non exhaustives
- Pose des problèmes de biais et de droits d'auteurs (les données utilisées pour l'apprentissage ne sont pas complètement connues)
- Pose des problèmes de confidentialité (les données utilisateurs sont utilisées par OpenAI)
- Pose des problèmes de fraude (appropriation de texte rédigé par autrui), non de plagiat (le texte généré par ChatGPT est généralement unique)

Exemples d'utilisation pour un élève

- S'inspirer du contenu généré par ChatGPT pour construire un devoir
- Résumer un cours
- Faire des traductions, des reformulations
- Poser des questions pour avoir des explications plus adaptées/précises sur des notions et du code informatique
- …

Exemples d'utilisation pour un (apprenti) développeur

- Obtenir des explications sur des concepts de programmation

  Ex de prompt : « Explique-moi le principe de l'héritage de la programmation objet »
- Générer des quizz pour tester ses connaissances

  Ex de prompt : « Génère-moi un quizz de 5 questions pour tester mes connaissances sur les LLM»
- Obtenir la documentation d'une fonction

  Ex de prompt : « Comment fonctionne la fonction plot de la librairie matplotlib ? »
- Obtenir des informations sur des éléments du langage

  Ex de prompt : « En Python, quelle est la différence entre les listes et les tuples ? »
- Générer du code pour faire un traitement particulier

  Ex de prompt : « Ecris le code PyTorch permettant de créer un CNN à trois couches de convolution et 2 couches denses»

  Ex de prompt : « Ecris le code pour entrainer ce modèle sur cifar10»

Exemples d'utilisation pour un (apprenti) développeur

- Générer des commentaires de code (dont docstring)

  Ex de prompt : « Réécris ce code Python en ajoutant les commentaires et les docstrings : [le code] »
- Détecter et expliquer des erreurs de codage

  Ex de prompt : « Quelle est l'erreur dans ce code Python : [le code] »
- Améliorer l'écriture / l'efficacité d'un code (temps, mémoire)

  Ex de prompt : « Réécris ce code en utilisant les recommandations PEP8 : [le code] »
- Générer les tests unitaires d'un code

  Ex de prompt : « Ecris moi les tests unitaires pour cette classe : [le code] »
- …

Conseils

- **Soigner le prompt :**
  - Donner le contexte
  - Être précis dans les instructions données
  - Affiner itérativement les instructions si besoin

- **Faire preuve d'esprit critique**
  - Le code généré ne doit jamais être utilisé sans vérification car il peut contenir des erreurs et/ou ne pas être adapté à la problématique (problème d'hallucination)

  ➔ ChatGPT ne remplacera pas le savoir-faire et l'expertise du programmeur, mais peut l'aider à gagner en efficacité

Alternatives à ChatGPT

CENTRALE LYON

Les alternatives

Nom	Fabricant	Open Source	Usage Commercial	Langues	Points Forts	Limites
ChatGPT	OpenAI	Non	Oui	Multilingue	Puissant (GPT-4), API bien documentée, plugins pour des tâches avancées	Modèle fermé, coûteux pour GPT-4 et usage intensif
Claude	Anthropic	Non	Oui	Multilingue	Axé sur la sécurité, alignement humain, réduction des biais	Moins personnalisable, coût élevé
Bard	Google	Non	Oui	Multilingue	Intégré dans Google Workspace, rapide et gratuit	Limitations dans les tâches complexes
LLaMA 2	Meta	Oui	Oui (sous licence)	Multilingue	Open source, personnalisable, performances élevées (7B à 70B)	Nécessite des ressources GPU importantes

Alternatives à ChatGPT

CENTRALE LYON

Les alternatives

Nom	Fabricant	Open Source	Usage Commercial	Langues	Points Forts	Limites
Falcon	Technology Innovation Institute (TII)	Oui	Oui	Multilingue	Très performant, coût d'exécution efficace, modèle compact	Configuration technique complexe pour les déploiements
BLOOM	BigScience	Oui	Oui	Multilingue	Inclusif et multilingue (175B), entièrement open source	Exige des ressources matérielles significatives
Jasper AI	Jasper	Non	Oui	Anglais principalement	Optimisé pour le marketing et la rédaction de contenu	Moins polyvalent pour des cas d'usage général
Bing Chat	Microsoft	Non	Oui	Multilingue	Alimenté par GPT-4, intégré au navigateur, gratuit	Limité en dehors de l'écosystème Microsoft
Mistral	Mistral AI	Oui	Oui	Multilingue	Compact (7B), performant pour sa taille, open source	Moins adapté aux tâches nécessitant de très grands modèles

Transformers & Images

Génération d'images : modèle Transformer adapté à la génération d'images

➔ apprend à associer des concepts textuels et visuels pour produire des images correspondant à des descriptions textuelles complexes

Nom	Open Source	Niveau Technique Requis	Qualité Visuelle	Public Cible
DALL·E	Non	Faible	Très élevée	Artistes, designers, créateurs de contenu
MidJourney	Non	Faible	Très élevée	Artistes, designers
Stable Diffusion	Oui	Élevé	Très élevée (personnalisée)	Développeurs, chercheurs
Imagen	Non	Moyen	Très élevée	Chercheurs, institutions
Runway ML (Gen-2)	Non	Faible	Moyenne à élevée	Créateurs multimédias
Artbreeder	Non	Faible	Moyenne à élevée	Amateurs, créateurs de profils
Craiyon	Oui	Faible	Moyenne	Débutants, explorateurs
DeepAI	Non	Faible	Moyenne	Débutants, généralistes
Jasper Art	Non	Faible	Moyenne à élevée	Marketeurs, créateurs de contenu

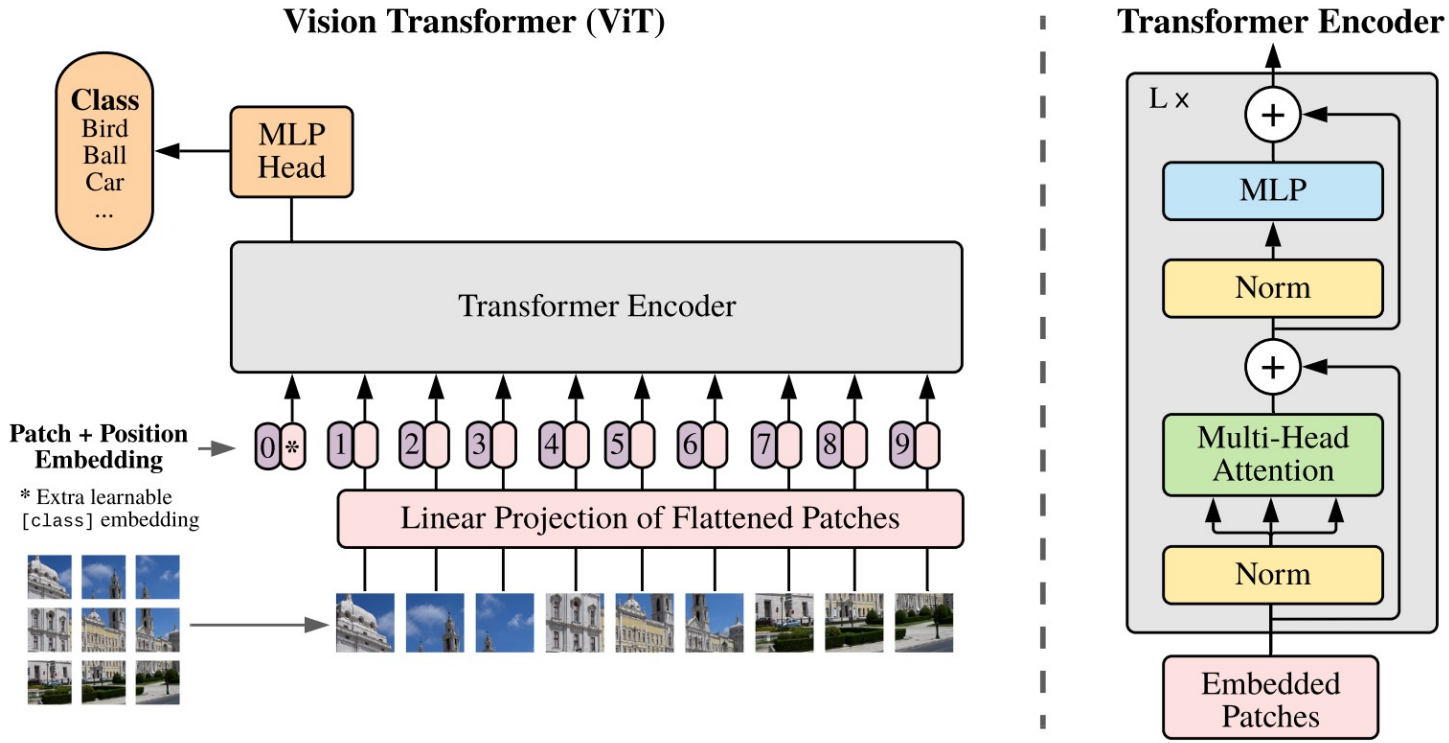
Transformers & Images

Classification d'images

Published as a conference paper at ICLR 2021

AN IMAGE IS WORTH 16X16 WORDS:
TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

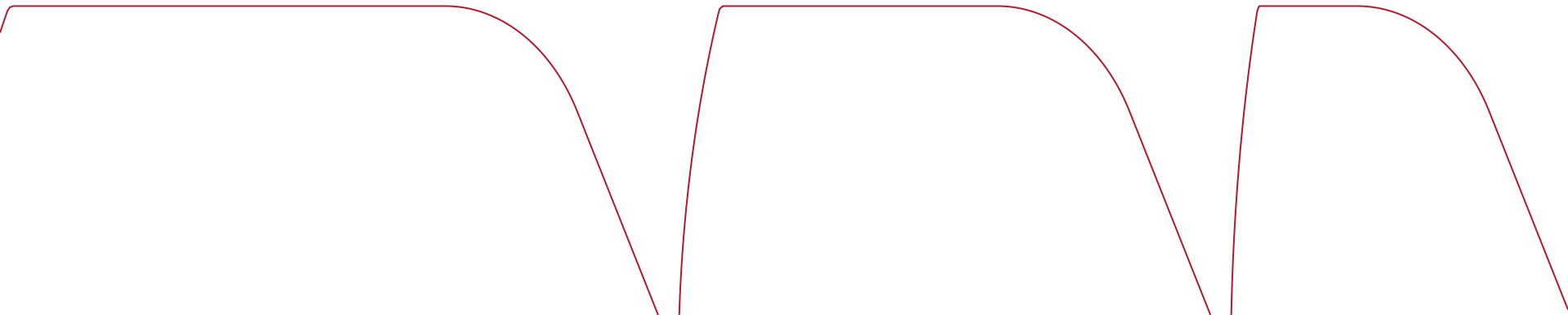
Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*,
Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†
*equal technical contribution, †equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com



Conclusion

- ChatGPT : outil très intéressant pour générer du texte souvent de très bonne qualité
- De nombreuses applications, et beaucoup encore à découvrir
- Ne pas se fier systématiquement aux réponses apportées ➔ faire preuve d'esprit critique
- Plusieurs outils en cours de développement pour essayer de détecter si un texte a été généré par ChatGPT (Compilatio, AI Text Classifier, DetectGPT, ZeroGPT, …)
- Forts impacts dans l'enseignement (pour les élèves et les enseignants) ➔ nécessité de réfléchir à son utilisation raisonnée

Exemple d’application de mise en œuvre d’un LLM avec Pytorch



LLM & Pytorch

CENTRALE LYON

- Installer PyTorch : https://pytorch.org/get-started/locally/
- et la librairie Hugging Face : https://huggingface.co/docs/transformers/llm_tutorial
- Puis tester :

```python
from transformers import AutoModelForCausalLM, AutoTokenizer

model = AutoModelForCausalLM.from_pretrained("gpt2")
tokenizer = AutoTokenizer.from_pretrained("gpt2")

prompt = "Suddenly, I saw"

input_ids = tokenizer(prompt, return_tensors="pt").input_ids

gen_tokens = model.generate(
    input_ids,
    do_sample=True,
    temperature=0.9,
    max_length=100,
)
gen_text = tokenizer.batch_decode(gen_tokens)[0]
print(gen_text)

```

Quelques références

PyTorch, https://pytorch.org/
Hugging Face, https://huggingface.co/
Open AI, https://openai.com/chatgpt/overview/
Formation FIDLE sur le Deep Learning, https://fidle.cnrs.fr/w3/

CENTRALE LYON

emmanuel.dellandrea@ec-lyon.fr

36, avenue Guy de Collongue 69130 Écully
www.ec-lyon.fr | @centralelyon