Partie 1 : Théorie des Transformers

Les Transformers sont devenus la référence en intelligence artificielle pour le traitement du langage naturel (NLP) et, plus récemment. Leur succès repose sur un concept clé : le mécanisme d'attention. Contrairement aux réseaux de neurones récurrents (RNN), ils permettent de traiter l'ensemble des données en parallèle, ce qui réduit le temps de calcul et améliore les performances.

1. Architecture d'un Transformer

Un Transformer est composé de plusieurs blocs qui se répètent. On distingue deux parties principales :

- **L'encodeur** : il transforme l'entrée en une représentation compréhensible pour le modèle. (Vectorisation)
- **Le décodeur** : il génère une sortie à partir de cette représentation (utile pour la traduction, par exemple).

Un modèle comme BERT utilise uniquement la partie encodeur, tandis qu'un modèle comme GPT exploite seulement le décodeur.

1.1 Le mécanisme d'attention

L'idée clé est de pondérer l'importance de chaque mot dans une phrase selon le contexte. Dans un Transformer, cette pondération est réalisée grâce à la fonction self-attention, qui permet au modèle d'établir des liens entre tous les mots d'une phrase en parallèle.

Le calcul de l'attention repose sur la formule suivante :

- **Q (Query)** : Représentation du mot qu'on analyse.
- **K (Key)** : Représentation des mots du contexte.
- **V (Value)** : Représentation sémantique associée.

Le softmax permet d'attribuer un poids à chaque mot en fonction de sa pertinence pour le mot analysé, mettant ainsi en avant les mots les plus significatifs pour une meilleure compréhension du texte.

Par exemple si on assiste à une discussion de groupe et que l'on essaye de comprendre ce que dit une personne. Plutôt que d'écouter uniquement celle-ci, on prête également attention aux réactions et aux interventions des autres, ce qui nous aide à mieux saisir le contexte global de la conversation. Le mécanisme d'attention fonctionne de manière similaire : il ne se concentre pas uniquement sur un mot, mais évalue aussi son interaction avec tous les autres mots de la phrase pour en tirer le sens le plus pertinent.

1.2 Attention multi-tête (multi Head attention)

Plutôt que d'utiliser une seule fonction d'attention, on en utilise plusieurs en parallèle (multi-head attention). Chaque tête d'attention apprend à repérer différents types de relations entre les mots d'une phrase, certaines se concentreront sur la structure grammaticale, tandis que d'autres capteront des relations de sens plus profondes.

L'idée est donc de permettre au modèle d'analyser un mot sous plusieurs angles différents et de combiner ces perspectives pour une meilleure compréhension du contexte.

Si par exemple on regarde un film et que plusieurs caméras filment différentes parties de la scène. Une caméra se concentre sur les visages des personnages, une autre sur leurs gestes, et une troisième sur le décor. Ensuite, toutes ces perspectives sont fusionnées pour obtenir une vision plus complète de la scène. C'est exactement ce que fait l'attention multi-tête avec le langage.

1.3 Positional Encoding

Les Transformers n'ont pas de notion de temps comme les RNN, ce qui signifie qu'ils ne comprennent pas naturellement l'ordre des mots dans une phrase. Pour résoudre ce problème, on ajoute une information de position à chaque mot sous forme de vecteurs sinusoïdaux.

Cette technique permet au modèle de savoir où se situe un mot par rapport aux autres et d'en tenir compte dans son analyse.

Par exemple si on assiste à une pièce de théâtre où les acteurs entrent et sortent de scène. Si on ne sait pas dans quel ordre ils apparaissent, l'histoire devient confuse. Le positional encoding joue alors le rôle d’un metteur en scène qui numérote chaque acteur pour garder le bon déroulement de l’histoire.

1.4 Couches Feedforward

Chaque bloc Transformer comprend une couche de transformation linéaire, aussi appelée MLP (Multi-Layer Perceptron), qui permet d'enrichir la représentation des tokens traités par l'attention.

Voici comment elle fonctionne :

1. **Première transformation linéaire** : Chaque vecteur issu du mécanisme d'attention passe par une couche dense qui projette les données dans un espace de plus grande dimension.
2. **Activation non linéaire (ReLU ou GELU)** : Cela permet d'introduire de la non-linéarité et d'augmenter la capacité du modèle à capturer des relations complexes.

3. **Deuxième transformation linéaire** : On projette les données dans un espace de dimension réduite pour revenir à la taille initiale des embeddings.
4. **Ajout et normalisation** : La sortie est ajoutée à l'entrée (skip connection), suivie d'une normalisation de couche (Layer Normalization) pour stabiliser l'entraînement.

Formellement, on a : Où **W_1**, **W_2** sont des matrices de poids et **b_1**, **b_2** des biais appris pendant l'entraînement.
Cette couche permet donc d'affiner et de complexifier la représentation des tokens après l'attention.

Plus concrètement l'attention sélectionne des mots importants dans une phrase. Mais parfois, ces informations brutes ne sont pas suffisantes. La couche feedforward agit alors comme un second filtre, transformant et affinant ces données avant de les envoyer à l'étape suivante. C'est comme un chef cuisinier qui goûte un plat (attention) et ajuste ensuite l'assaisonnement (feedforward) pour améliorer le goût global.

2. Variantes des Transformers

2.1 BERT : Bidirectional Encoder Representations from Transformers

BERT est un modèle basé uniquement sur l'encodeur. Il est entraîné avec deux méthodologies :

- **Masked Language Modeling (MLM)** : On masque des mots d'une phrase et le modèle doit les deviner.
- **Next Sentence Prediction (NSP)** : Le modèle prédit si une phrase suit logiquement une autre.

Bert est très performant dans les taches suivantes : Classification de texte, recherche d'information, analyse de sentiments, etc.

2.2 Vision Transformer (ViT)

L'idée de ViT est d'appliquer les Transformers aux images. Plutôt que de considérer chaque pixel individuellement, on découpe l'image en **patches** (petites régions carrées de pixels) et on les traite comme des tokens.

Le processus est le suivant :

1. **Découpage de l'image** en patches de taille fixe.
2. **Transformation de chaque patch** en un vecteur numérique (embedding) grâce à une couche linéaire.
3. **Traitement par un Transformer** : chaque vecteur est analysé avec le mécanisme d'attention pour extraire des informations pertinentes et classifier l'image.

Avantages :

- Réduction du besoin en calculs par rapport à un réseau convolutionnel (CNN) profond.
- Capture de relations globales dans l'image, contrairement aux CNN qui se focalisent sur des détails locaux.

Si on imagine une image comme un puzzle. Chaque pièce (patch) est analysée individuellement avant d'être assemblée pour comprendre l'image complète. Contrairement aux CNN qui examinent l'image de près en repérant les petits motifs, ViT regarde l'image dans son ensemble pour mieux en saisir la structure globale.

Pour conclure les Transformers ont révolutionné l'IA en permettant un traitement plus efficace des données textuelles et visuelles. Ils exploitent le self-attention pour capturer les relations complexes entre les éléments d'une séquence. BERT excelle en NLP, tandis que ViT montre que ces modèles peuvent aussi surpasser les CNN pour la vision.