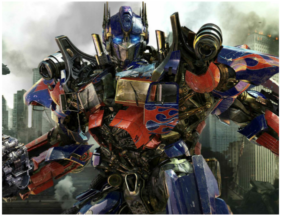


TRANSFORMERS

Vincent Barra
LIMOS, UMR 6158 CNRS, Université Clermont Auvergne

INTRODUCTION

Why ?

- Transformers have revolutionized NLP
- Emerged as a powerful paradigm for computer vision applications, imaging or multi-modal datasets.

Key points

- Attention mechanism
- Self attention

LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

UNIVERSITÉ Clermont Auvergne

Basic form of self attention

Self attention philosophy

Convert one set of input embeddings to a set of output embeddings.

Input sequence: T vectors $\mathbf{x}_1 \cdots \mathbf{x}_T$ of size $d_e \to \mathbf{X} \in \mathcal{M}_{T,d_e}(\mathbb{R})$

$$\mathbf{x}_i = \begin{cases} \text{timeseries} \\ \text{flattened part of an image} \\ \text{tokens representing words (LLM)} \\ \dots \end{cases}$$

Embedding: related inputs are represented by similar vectors

Self-attention mechanism transforms $\mathbf{x}_1 \cdots \mathbf{x}_T$ to $\mathbf{y}_1 \cdots \mathbf{y}_T$ where $\mathbf{y}_i = \sum_{j=1}^{T} W_{ij}\mathbf{x}_j$

Goal

Create context-aware output vectors, i.e. output vectors that account for the pair-wise relationships between input vectors

LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

UNIVERSITÉ Clermont Auvergne

EXAMPLE

"The Queen died on September 8^{th} 2022"

"He uses the Queen to beat his opponent in the match"

$\mathbf{x}_i = "Queen"$

$\Rightarrow$ The transformation $\mathbf{y}_i = \sum_{j=1}^{T} W_{ij}\mathbf{x}_j$ aims to create representations that are aware of the surrounding words, and thus the context the word is used in

LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

UNIVERSITÉ Clermont Auvergne

WEIGHT MATRIX

Weights

- The weight matrix is derived from the inputs
- High weight to those inputs that have a high similarity to $\mathbf{x}_i$.
- Example= $w_{ij} = \mathbf{x}_i^T \mathbf{x}_j$

We want positive weights summing up to one $\Rightarrow$ softmax

$\rightarrow$ $w_{ij} = exp(w_{ij})$

$\rightarrow$ $W_{ij} = \dfrac{w_{ij}}{\sum_j w_{ij}}$

$\mathbf{x}_1 \cdots \mathbf{x}_T$ have been transformed into weighted combinations of all the other vectors, with weights being larger for input vectors that are more similar (it "attends" more to them).

LIMOS LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

cnrs

UCA UNIVERSITÉ Clermont Auvergne

Learnable weights

For the moment, weights are not learnable $\Rightarrow$ the model cannot be optimized for a specific task.

$$\mathbf{y}_i = \sum_j softmax(\mathbf{x}_i^T \mathbf{x}_j)\mathbf{x}_j$$

$\mathbf{x}_i$ is used three times in three different roles
$\Rightarrow$ modify each occurrence of $\mathbf{x}_i$ by multiplying it with a different matrix containing learnable weights.

Query, Key, and Value

$$\mathbf{q}_i = \mathbf{x}_i \mathbf{W_Q}$$
$$\mathbf{k}_i = \mathbf{x}_i \mathbf{W_K}$$
$$\mathbf{v}_i = \mathbf{x}_i \mathbf{W_V}$$

LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

Query, Key, and Value??

- Suppose you want to find a YouTube video on how to build a house.
- You write on the search tab "How to build a house" $\rightarrow$ **Query**
- Every YouTube video has some keywords related to its contents $\rightarrow$ **Key**
- The similarity of the Query to the Keys of each video is computed
- Videos with the highest similarity scores are returned at the top of the search.
- The content of these top-ranked videos contains the **Value** (information) that best answers the initial question (Query)

Attention mechanism

Each input vector acts as a Query that is compared to every other vector (Keys) to find those combinations with the highest similarity (W_{ij}).
The output is a linear combination of the input, dominated by those input vectors (Values) with the highest attention scores.

LIMOS LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

UNIVERSITÉ Clermont Auvergne

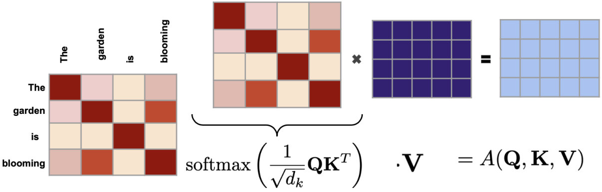
Query, Key, and Value

Query, Key, and Value

$\mathbf{q}_i = \mathbf{x}_i \mathbf{W_Q}, \mathbf{W_Q} \in \mathcal{M}_{d_e, d_q}(\mathbb{R})$
$\mathbf{k}_i = \mathbf{x}_i \mathbf{W_K}, \mathbf{W_K} \in \mathcal{M}_{d_e, d_k}(\mathbb{R})$
$\mathbf{v}_i = \mathbf{x}_i \mathbf{W_V}, \mathbf{W_V} \in \mathcal{M}_{d_e, d_v}(\mathbb{R})$
$\mathbf{W_Q}, \mathbf{W_K}, \mathbf{W_V}$: projection matrices on lower dimensional spaces

$$W_{ij} = softmax\left(\frac{\mathbf{q}_i \mathbf{k}_j^T}{\sqrt{d_k}}\right)$$

$$\mathbf{y}_i = \sum_{j=1}^{T} W_{ij} \mathbf{v}_j \rightarrow Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{1}{\sqrt{d_k}} \mathbf{Q}\mathbf{K}^T\right) \mathbf{V}$$



Summary

Up to now

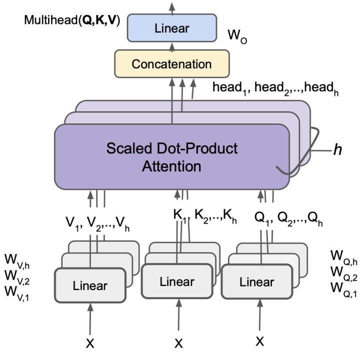
- the similarity matrix $\mathbf{QK}^T$ is used to create the attention weights that capture the relevance of the combination of inputs
- these attention weight matrices multiply the matrix of value vectors to extract important features that are subsequently passed to the next stage of the network.
- $\mathbf{W_Q}, \mathbf{W_K}, \mathbf{W_V}$: attention head.

LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

Multi-head attention

Multi-head attention

- ▶ CNN: multiple kernels
- ≈ multiple $\mathbf{W_Q}, \mathbf{W_K}, \mathbf{W_V}$: capture different relationships between the input embeddings.
- ▶ h attention heads in parallel
- ▶ $\mathbf{W}^O$: linear projection matrix used in order to project the concatenated vector to the output model space



$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat(h_1, \cdots h_h)\mathbf{W}^O, \; h_i = Attention(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)$$

The $T \times d_e$ input is transformed into $T \times d_O$

LIMOS LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

cnrs

UCA UNIVERSITÉ Clermont Auvergne

Positional encoding

As described for now, the self-attention mechanism is oblivious to the position of the inputs.
$\Rightarrow$ Architecture invariant under permutations.

The position is important (image, words,...)

Positional encoding: $\mathbf{p}_i \in \mathbb{R}^{d_e}$

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{p}_i$$

i: position in the sequence.

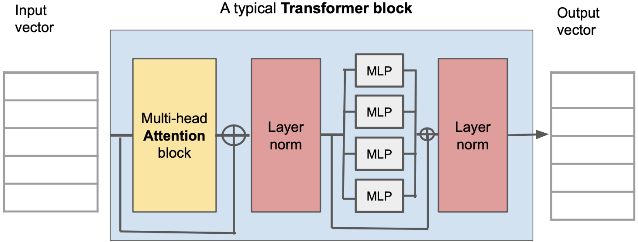
- learnable $\mathbf{p}_i$: positional embedding
- specific form $\mathbf{p}_i$: encodes the position

Example (seminal work)

$$\left\{ \begin{array}{ccc} \mathbf{p}_i(2j+1) & = & cos\left(\frac{1}{10000^{2j/d_e}}\right) \\ \mathbf{p}_i(2j) & = & sin\left(\frac{1}{10000^{2j/d_e}}\right) \end{array} \right.$$

The transformer block

Transformer: any architecture that has a transformer block as a basic building element.

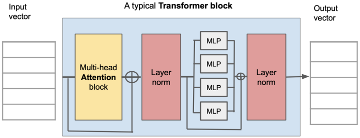


LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

The transformer block

Multi-head attention

- Skip connections: $\mathbf{X}' = MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + X$
- Layer norm: normalize w.r.t. mean and standard deviations of the hidden units of the layer
- MLP: Increase the capability of the model without increasing its computational complexity (MLP acts on each position independently, the attention mechanism having already learned the correlations across positions)



LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

UNIVERSITÉ Clermont Auvergne