TRANSFORMERS : heralding a new era in sequence processing

Saptarshi Datta

as part of CS-460 Machine Learning course guided by Dr. Subhankar Mishra

Introduction

- The new model proposed in the paper "Attention is All You Need" [1] makes use of a previously uncharted territory and showed remarkable success in sequence processing.
- The primary focus was on developing language model and the Transformer showed significantly better results than previously used RNNs or CNNs.
- Single-head Self Attention was a known concept previously. The main development was by introducing Multi-head Self Attention.
- A Transformer Model has two major parts the Encoder and the Decoder.

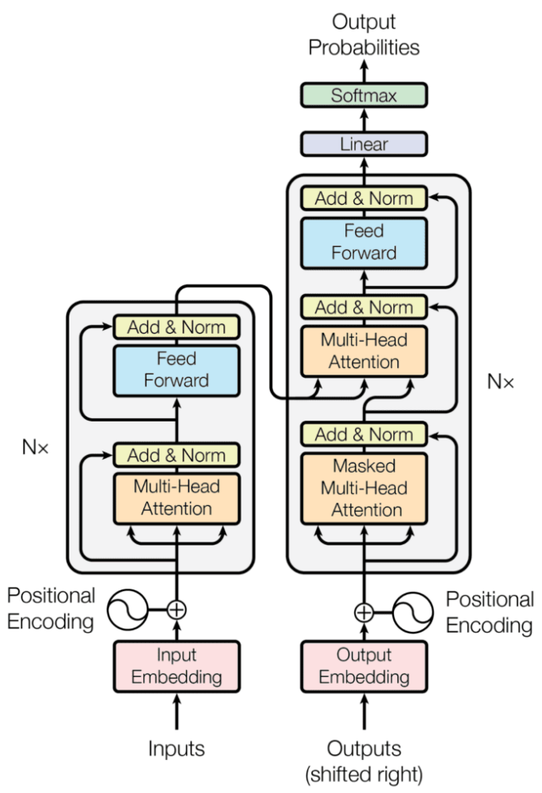

Fig 1. Schematic of a Transformer Architecture [1]

Encoder-Decoder Architecture

The encoder consists of a stack of N=6 identical layers, where each layer is composed of two sub-layers:

1) The first sub-layer implements a multi-head self-attention mechanism. It implements ***h*** heads that receive a (different) linearly projected version of the queries, keys, and values, each to produce ***h*** outputs in parallel that are then used to generate a final result.

2) The second sub-layer is a fully connected feed-forward network consisting of two linear transformations with Rectified Linear Unit (ReLU) activation in between: $\mathrm{FFN}(x) = \mathrm{ReLU}\,(\mathbf{W}_1 x + b_1)\,\mathbf{W}_2 + b_2$

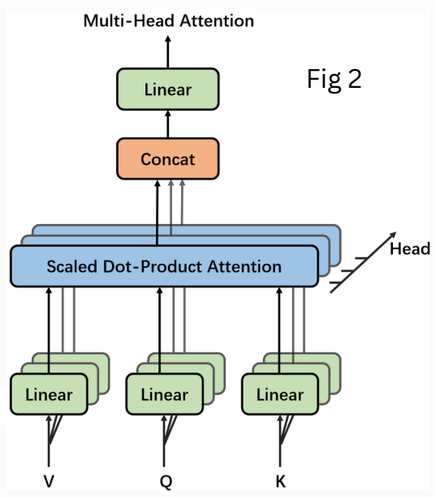

Fig 2

Each sub-layer is also succeeded by a normalisation layer, *layernorm(.)*, which normalises the sum computed between the sub-layer input, *x*, and the output generated by the sub-layer itself, *sublayer(x)*:

Normalised = *layernorm* (*x* + *sublayer(x)*)

The Transformer architecture cannot inherently capture any information about the relative positions of the words in the sequence since it does not make use of recurrence. This information has to be injected by introducing positional encoding to the input embedding.

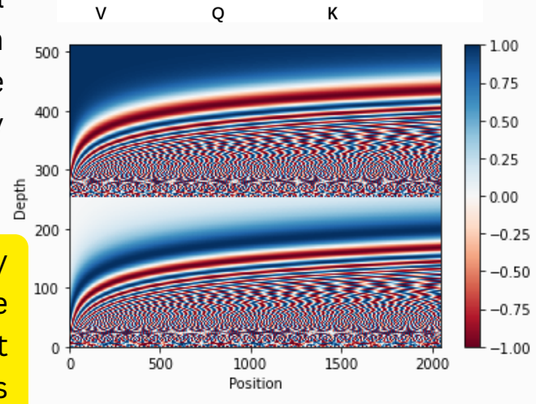

Fig 3. Positional encoding as represented in [1]

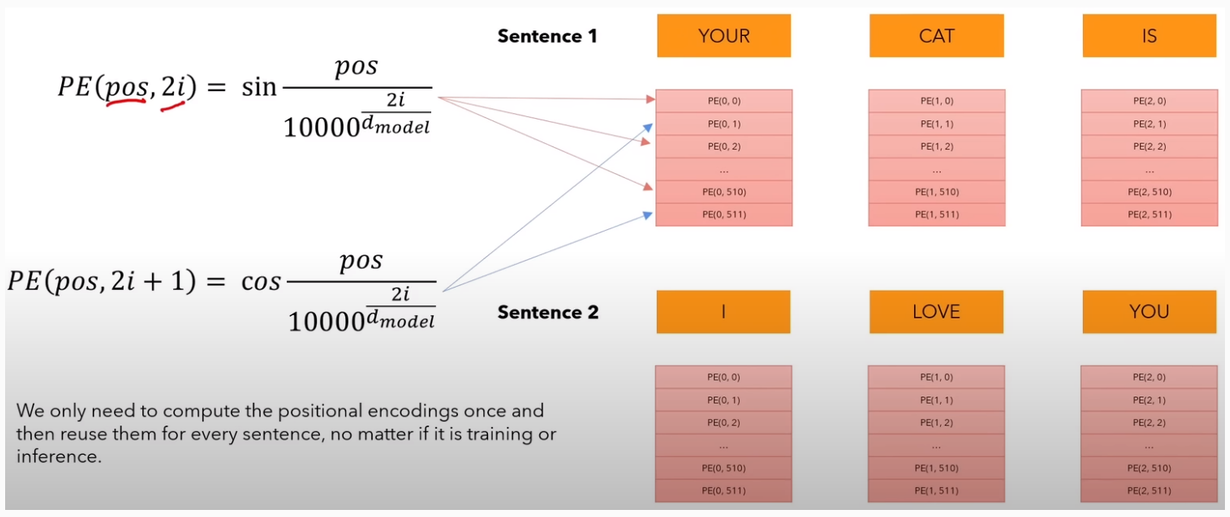

Fig 4. A practical example of positional encoding

Masking ensures that the correlation (matrix components) between one word to its successive words is made 0 while training. This is to ensure that the model does not learn with a prediction bias. Taking the *softmax(.)* on this matrix solves it.

The multi-head mechanism receives the queries from the previous decoder sub-layer and the keys and values from the output of the encoder. This allows the decoder to attend to all the words in the input sequence. The third layer implements a fully connected feed-forward network, similar to the one implemented in the second sub-layer of the encoder.

Fig 5. Masking of successive words followed by Multi-Head attention

Training and Inference

- For training purposes, the input sequence is prefix appended with a <START> token and suffix appended with a <END> token. The same is done for the output if it is also a sequence.
- The encoder generates the ENCODED matrix which includes correlation between each elements in the sequence as well as positional encoding. This is provided as keys and values to the second decoder sub-layer while the query comes from the Masked Multi-Head Attention (first sub-layer) of the decoder.
- Upon correct training of the model, the Output Probabilities should cross a certain threshold (and may not improve significantly any further).

Inference is the implementation of a trained model. Once we have a trained Transformer model, and assuming that the output is also a sequence, we supply the <START> token to the Output and the input is as given. The ENCODED matrix once calculated remains same for the input and we successively modify the tokens supplied to the Decoder module. This continues till we receive the <END> token as the output.

Why use TRANSFORMER?

- This is primarily Due to the self-attention mechanism, Transformers can process all elements of a sequence in parallel unlike RNNs. It enables faster training and inference times, as multiple computations can be performed simultaneously across different parts of the input sequence.
- It allows Transformers to take advantage of hardware accelerators like GPUs and TPUs more effectively, leading to significant speedups in training and inference compared to sequential models.
- Transformers are highly scalable, meaning they can handle input sequences of varying lengths without a significant increase in computational complexity. Unlike recurrent models, which suffer from vanishing or exploding gradients when processing long sequences, Transformers maintain stable gradients throughout the entire sequence length.
- Transformers can be scaled up in terms of model size (number of layers, hidden units, etc.) to improve performance on complex tasks or larger datasets without sacrificing efficiency. With advancements like sparse attention mechanisms and model pruning techniques, researchers are further enhancing the scalability of Transformers, enabling them to handle even larger datasets and more complex tasks.

Applications

- Text Generation: Models like GPT (Generative Pre-trained Transformer) are proficient in generating coherent and contextually relevant text.
- Sentiment Analysis: Transformers-based models effectively analyse sentiment in text data, facilitating applications in social media monitoring, customer feedback analysis, and opinion mining.
- Question Answering: Transformer-based architectures, such as BERT (Bidirectional Encoder Representations from Transformers), are capable of answering questions based on context provided in the input text.
- Named Entity Recognition (NER): Transformers are adept at identifying and classifying named entities (e.g., person names, locations, organizations) in text data, which is useful in various information extraction tasks.
- Language Understanding: Transformers enable advanced natural language understanding capabilities, facilitating applications like intent detection, dialogue systems, and virtual assistants.
- Speech Recognition: Transformers can be adapted for speech recognition tasks, where they process audio input to transcribe spoken words into text, enabling applications in voice-controlled interfaces and dictation software.

References

1. “Attention Is All You Need”. Vaswani, A et al. https://doi.org/10.48550/arXiv.1706.03762
2. https://en.wikipedia.org/wiki/Transformer_(deep_learning_architecture)
3. https://machinelearningmastery.com/the-transformer-model/
4. https://machinelearningmastery.com/the-transformer-attention-mechanism/
5. Various open source websites and YouTube

Acknowledgement

- **Fellowship :** Dept. of Atomic Energy, Govt. of India
- **Valuable feedback and comments :** Dr. Subhankar Mishra, Aniket Nath (TA, CS460), batch-mates undertaking this course