Note on Transformer: "Attention Is All You Need"

Created on 2024-2-12
Author Di Yu (yudi.0211@foxmail.com)

Introduction

The Transformer is a machine learning model for sequence transduction, featuring high prediction accuracy, low computational complexity, and high computation parallelization. This model is the building block of modern large language models like GPT-3, which is conceived as a competitive candidate for realizing artificial general intelligence (AGI). The Transformer was firstly proposed in NIPS 2017 by a research group at Google. The associated paper titled "Attention Is All You Need" [1] has been cited for over 100,000 times as of 2024, demonstrating the huge impact of the Transformer on NLP studies. In this note, I will introduce the architecture of the Transformer and share my understanding of why it works.

Background

Existing neural sequence transduction models based on recurrent neural networks (RNNs) and attention mechanism exhibited state-of-the-art performance on machine translation tasks [2]. However, the inherent sequential property of RNNs makes it difficut to accelerate the training process with parallel computing. This makes the training process time consuming and hinders the development of large-scale NLP models.

Architecture of the Transformer

The basic architecture of the Transformer is illustrated in the figure below. The Transformer employs an encoder-decoder architecture, like many previous neural sequence transduction models. The encoder consists of 6 identical layers, each comprises 2 sublayers: a self-attention function and a feed-forward neural network. The decoder also consists of 6 identical layers, but each layer includes an additional self-attention function. Residual connections are used in both the encoder and the decoder to improve training accuracy [3]. The input text is firstly converted into 512-dimensional vectors by an embedding a layer. Positional encoding is then applied to the vector to inject text sequential information into the model, prior to further processing by attention functions and neural networks.

Multi-head attention

The Transformer utilizes multi-head attention function to identify the correlation within a sequence and between two different sequences. Given an input vector (aka query) Q and a dictionary consisting of key-value pairs (K, V), the multi-head attention function identifies their correlation by calculating the dot product between the input vector and all keys, namely QK^T. Note that all keys and values in the dictionary are 512-dimensional vectors, which ensures the dot product described here to be legitimate. The output of the attention function is a weighted average of the values in the dictionary, and the weight coefficients are softmax function of the dot product.

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

Here, the scaling factor $\sqrt{d_k}$ serves to counter the increase in dot product as vector dimension grows, which might lead to diminishing gradients and degrading training accuracy.

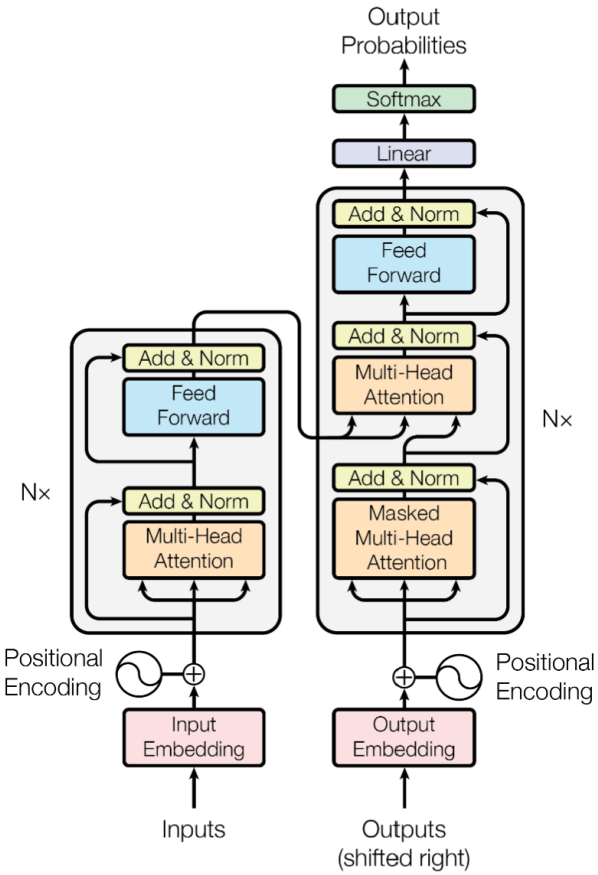

Figure 1: Architecture of the Transformer

It's easy to see that the attention function employs dot product to calculate the correlation between two sequences, which implicitly hypothesizes equal treatment of all kinds of correlation. In practice, some correlations are more important than others in predicting the model output. To equip the Transformer with this capability of discrimination, in practice, the attention function first projects the query vector and all keys and values to several subspaces, and attention functions are evaluated in each subspace individually. The calculated attention in each subspace (single-head attention) is concatenated to form the final output of the multi-head attention function. Note that the projection parameters are to be optimized, which enables the Transformer to self-adaptively judge the relative importance of different kinds of correlation.

The multi-head attention function can be calculated efficiently using existing matrix multiplication algorithms, which feature relatively low computational complexity. Moreover, the vector nature of the multi-head attention function makes it suitable for parallel computing acceleration. These two advantages of the multi-head attention function make it a favorable option compared to conventional recurrent neural network-based attention mechanisms.

Feed-forward neural networks

Each layer of the encoder and the decoder comprises a fully connected feed-forward neural network sublayer that follows a multi-head attention sublayer. This feed-forward neural network sublayer consists of two linear transformations with a ReLU activation function. The hidden layer has a dimension of 2048.

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2.$$

This feed-forward neural network provides nonlinear activation functions for the model, which is otherwise absent considering the construction of the multi-head attention function.

Observations, Comments, and Interesting Facts

- All authors are marked with 'equal contribution'.
- The Transformer model established a new SOTA result on WMT 2014 English-to-French translation task after training for 3.5 days on 8 P100 GPUs.
- The Transformer has two main advantages: it's compatible with parallel computing and has low computational complexity, establishing itself as a promising building block for future large language models.
- This paper [1] proposes the multi-head attention function to suprecede recurrent neural network-based attention function, which features high computation cost due to its inherent sequential nature.
- While ResNet was firstly proposed to improve the performance of deep learning models for dealing with computer vision tasks, here it is incorporated in an NLP model.
- The Transfomer uses a combination of a multi-head attention function and a fully connected feed-forward neural network to calcualte a counterpart of weighted hidden states, an essential component for attention mechanism [2].
- The Transformer employs a positional encoding to inject sequential information to the model. This encoding adds an offset to the embedded vectors depending on the position of the text. Importantly, the positional encoding injects sequential information without breaking the model's compatibility with parallel computing.

Reference

---

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

2. Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
3. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).