transformer

Authors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Abstract

A transformer is an artificial neural network (ANN) that is built by composing layers, each of which transforms the features of a data point that consists of tokens, such as the words of a sentence. The feature vectors of the tokens are stacked into a matrix, and each layer maps this matrix to a transformed matrix of the same shape, so that layers can be composed freely into a deep ANN. The layers alternate between token mixing, which combines information across tokens, and a position-wise multilayer perceptron (MLP), which transforms each token separately. In the original transformer, token mixing is implemented by multi-head attention. Alternative mixing transformations, such as a fixed Fourier transform or a state-space layer, can take the place of attention. Transformers underlie large language models (LLMs).

Definition

In the context of machine learning (ML), the term transformer refers to an artificial neural network (ANN) that is built by composing layers, each of which implements a transformation of the features of a data point that consists of tokens (Vaswani et al., 2017). The feature vectors (or embeddings) $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)} \in \mathbb{R}^d$ of the n tokens are stacked into a matrix $\mathbf{X} = \big(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\big)^\top \in \mathbb{R}^{n \times d}$. Each layer of a transformer maps such a matrix to a transformed matrix of the same shape. Since input and output have the same shape, the layers can be composed freely into a deep ANN. For example, a transformer that translates a sentence transforms the stacked word embeddings layer by layer into feature vectors from which the translated words are predicted (Vaswani et al., 2017); the same principle underlies large language models (LLMs).

The layers of a transformer alternate between two types of transformations of $\mathbf{X}$ (see Fig. 1). A token-mixing layer combines information across tokens, i.e., across the rows of $\mathbf{X}$. In the original transformer, token mixing is implemented by multi-head attention; the attention mechanism is what sets transformers apart from previous models for sequential data such as recurrent neural networks (RNNs). A position-wise multilayer perceptron (MLP) then transforms each row of $\mathbf{X}$ separately, so that tokens interact only through the token-mixing layers. Residual connections and layer normalization are interleaved with these transformations.

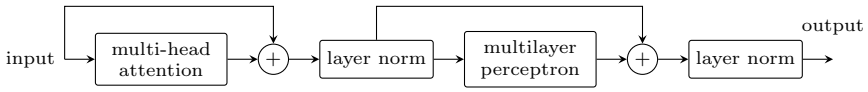

Figure 1: Signal-flow chart of one transformer block. The input is processed in parallel by a multi-head attention layer (the token-mixing layer) and forwarded by a residual connection to the first addition; layer normalization, a position-wise multilayer perceptron, and a second residual + normalization follow. A transformer ANN stacks many such blocks

Token mixing need not be implemented by attention. Any transformation that combines the rows of $\mathbf{X}$ can take its place: a multilayer perceptron applied across tokens (Tolstikhin et al., 2021), a fixed Fourier transform (Lee-Thorp et al., 2022), or a state-space layer whose cost grows only linearly in the number n of tokens (Gu and Dao, 2024). These alternatives replace the pairwise token comparisons of attention, whose cost grows as n^2, with cheaper mixing transformations (see attention).

Cross References

attention, ANN, layer, token, embedding, RNN, NLP, LLM, multilayer perceptron

References

1. Gu and Dao (2024). Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In: Conf. Language Modeling.

2. Lee-Thorp et al. (2022). FNet: Mixing Tokens with Fourier Transforms. In: Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics.

3. Tolstikhin et al. (2021). MLP-Mixer: An all-MLP Architecture for Vision. In: Adv. Neural Inf. Process. Syst.

4. Vaswani et al. (2017). Attention is all you need. In: Adv. Neural Inf. Process. Syst., pp. 5998–6008.