Transformers in Machine Learning

Last Updated : 27 Feb, 2025

Transformer is a neural network architecture used for performing machine learning tasks particularly in natural language processing (NLP) and computer vision. In 2017 Vaswani et al. published a paper *" Attention is All You Need"* in which the transformers architecture was introduced. The article explores the architecture, workings and applications of transformers.

Need For Transformers Model in Machine Learning

Transformer Architecture is a model that uses **self-attention** to transform one whole sentence into a single sentence. This is useful where older models work step by step and it helps overcome the challenges seen in models like RNNs and LSTMs. Traditional models like RNNs (Recurrent Neural Networks) suffer from the **vanishing gradient** problem which leads to long-term memory loss. RNNs process text sequentially meaning they analyze words one at a time.

> *For example, in the sentence:* ***"XYZ went to France in 2019 when there were no cases of COVID and there he met the president of that country"*** *the word "****that country****" refers to "****France****".*
>
> *However RNN would struggle to link "that country" to "France" since it processes each word in sequence leading to losing context over long sentences. This limitation prevents RNNs from understanding the full meaning of the sentence.*

While adding more memory cells in LSTMs (Long Short-Term Memory networks) helped address the vanishing gradient issue they still process words one by one. This sequential processing means LSTMs can't analyze an entire sentence at once.

For instance the word "point" has different meanings in these two sentences:

- "The needle has a sharp point." (Point = Tip)
- "It is not polite to point at people." (Point = Gesture)

Traditional models struggle with this context dependence, whereas, **Transformer model through its self-attention mechanism, processes the entire sentence in parallel addressing these issues and making it significantly more effective at understanding context.**

Architecture and Working of Transformers

1. Positional Encoding

Unlike RNNs transformers lack an inherent understanding of word order since they process data in parallel. To solve this Positional Encodings are added to token embeddings providing information about the position of each token within a sequence.

2. Position-wise Feed-Forward Networks

The Feed-Forward Networks consist of two linear transformations with a ReLU activation. It is applied independently to each position in the sequence.

Mathematically:

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

This transformation helps refine the encoded representation at each position.

3. Attention Mechanism

The attention mechanism allows transformers to determine **which words in a sentence are most relevant** to each other. This is done using a **scaled dot-product attention** approach:

1. Each word in a sequence is **mapped to three vectors**:

- **Query (Q)**
- **Key (K)**
- **Value (V)**

2. Attention scores are computed as: $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$

3. These scores determine how much attention each word should pay to others.

Multi-Head Attention

Instead of using a single attention mechanism transformers apply **multi-head attention** where multiple attention layers run in parallel. This enables the model to **capture different types of relationships within the input**.

4. Encoder-Decoder Architecture

The encoder-decoder structure is key to transformer models. The encoder processes the input sequence into a vector, while the decoder converts this vector back into a sequence. Each encoder and decoder layer includes **self-attention** and **feed-forward layers**. In the decoder, an encoder-decoder attention layer is added to focus on relevant parts of the input.

> *For example, a French sentence **"Je suis étudiant"** is translated into **"I am a student"** in English.*

The **encoder** consists of multiple layers (typically **6 layers**). Each layer has **two main components**:

- **Self-Attention Mechanism** – Helps the model understand word relationships.
- **Feed-Forward Neural Network** – Further transforms the representation.

The **decoder** also consists of **6 layers**, but with an additional **encoder-decoder attention** mechanism. This allows the decoder to focus on **relevant parts of the input sentence** while generating output.

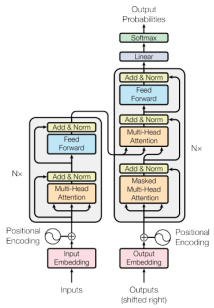

Figure 1: The Transformer - model architecture.

For instance in the sentence ***"The cat didn't chase the mouse, because it was not hungry",*** the word 'it' refers to 'cat'. The self-attention mechanism helps the model correctly associate 'it' with 'cat' ensuring an accurate understanding of sentence structure.

Applications of Transformers

Some of the applications of transformers are:

1. **NLP Tasks**: Transformers are used for machine translation, text summarization, named entity recognition and sentiment analysis.
2. **Speech Recognition**: They process audio signals to convert speech into transcribed text.
3. **Computer Vision**: Transformers are applied to image classification, object detection, and image generation.
4. **Recommendation Systems**: They provide personalized recommendations based on user preferences.
5. **Text and Music Generation**: Transformers are used for generating text (e.g., articles) and composing music.

Transformers have redefined deep learning across NLP, computer vision, and beyond. With advancements like BERT, GPT and Vision Transformers (ViTs) they continue to push the boundaries of AI and language understanding and multimodal learning.