From Wikipedia, the free encyclopedia

An illustration of main components of the transformer model from the paper "Attention Is All You Need"[1] is a 2017 landmark[2][3] research paper in machine learning authored by eight scientists working at Google. The paper introduced a new deep learning architecture known as the transformer, based on the attention mechanism proposed in 2014 by Bahdanau et al.[4] It is considered a foundational[5] paper in modern artificial intelligence, and a main contributor to the AI boom, as the transformer approach has become the main architecture of a wide variety of AI, such as large language models.[6][7] At the time, the focus of the research was on improving Seq2seq techniques for machine translation, but the authors go further in the paper, foreseeing the technique's potential for other tasks like question answering and what is now known as multimodal generative AI.[1]

The paper's title is a reference to the song "All You Need Is Love" by the Beatles.[8] The name "Transformer" was picked because Jakob Uszkoreit, one of the paper's authors, liked the sound of that word.[9]

An early design document was titled "Transformers: Iterative Self-Attention and Processing for Various Tasks", and included an illustration of six characters from the Transformers franchise. The team was named Team Transformer.[8]

Some early examples that the team tried their Transformer architecture on included English-to-German translation, generating Wikipedia articles on "The Transformer", and parsing. These convinced the team that the Transformer is a general purpose language model, and not just good for translation.[9]

As of 2025, the paper has been cited more than 173,000 times,[10] placing it among top ten most-cited papers of the 21st century.[11]

Authors

The authors of the paper are: Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones [wikidata], Aidan Gomez, Å•ukasz Kaiser, and Illia Polosukhin. All eight authors were "equal contributors" to the paper; the listed order was randomized. The Wired article highlights the group's diversity:[8]

Six of the eight authors were born outside the United States; the other two are children of two green-card-carrying Germans who were temporarily in California and a first-generation American whose family had fled persecution, respectively.

After the paper, each of the authors left Google to join other companies or to found startups. Several of them expressed feelings of being unable to innovate and expand the Transformer in a direction they want, if they had stayed at Google.[12]

Methods discussed and introduced

The paper is most well known for the introduction of the Transformer architecture, which forms the underlying architecture for most forms of modern Large Language Models (LLMs). A key reason for why the architecture is preferred by most modern LLMs is the parallelizability of the architecture over its predecessors. This ensures that the operations necessary for training can be accelerated on a GPU allowing both faster training times and models of bigger sizes to be trained.

The following mechanisms were introduced by the paper as part of the development of the transformer architecture.

Scaled dot-product attention & self-attention

The use of the scaled dot-product attention and self-attention mechanism instead of a Recurrent neural network or Long short-term memory (which rely on recurrence instead) allow for better performance as described in the following paragraph. The paper described the scaled dot-product attention as follows:

Attention(Q, K, V) = softmax((QÂ·Káµ€) / âˆšd_k) Â· V

where Q, K, V are respectively the query, key, value matrices, and d_k is the dimension of the values.

Since the model relies on Query (Q), Key (K) and Value (V) matrices that come from the same source itself (i.e. the input sequence / context window), this eliminates the need for RNNs completely ensuring parallelizability for the architecture. This differs from the original form of the Attention mechanism introduced in 2014. Additionally, the paper also discusses the use of an additional scaling factor that was found to be most effective with respect to the dimension of the key vectors (represented as d_k and initially set to 64 within the paper) in the manner shown above.

In the specific context of translation which the paper focused on, the Query and Key matrices are usually represented in embeddings corresponding to the source language while the Value matrix corresponds to the target language.

Multi-head attention

In the self-attention mechanism, queries (Q), keys (K), and values (V) are dynamically generated for each input sequence (limited typically by the size of the context window), allowing the model to focus on different parts of the input sequence at different steps. Multi-head attention enhances this process by introducing multiple parallel attention heads. Each attention head learns different linear projections of the Q, K, and V matrices. This allows the model to capture different aspects of the relationships between words in the sequence simultaneously, rather than focusing on a single aspect.

By doing this, multi-head attention ensures that the input embeddings are updated from a more varied and diverse set of perspectives. After the attention outputs from all heads are calculated, they are concatenated and passed through a final linear transformation to generate the output.

Positional encoding

Since the Transformer model is not a seq2seq model and does not rely on the sequence of the text in order to perform encoding and decoding, the paper relied on the use of sine and cosine wave functions to encode the position of the token into the embedding. The methods introduced in the paper are discussed below:

PE(pos, 2i)   = sin(pos / 10000^(2i/d_model))
PE(pos, 2i+1) = cos(pos / 10000^(2i/d_model))

wherein pos, i, d_model correspond to the position of the word, the current dimension index and the dimension of the model respectively. The sine function is used for even indices of the embedding while the cosine function is used for odd indices. The resultant PE embedding is then added to the word at that corresponding position with respect to the current context window. The paper specifically comments on why this method was chosen describing:

"We chose the sinusoidal version because it may allow the model to extrapolate to sequence lengths longer than the ones encountered during training."[1]

Historical context
Main articles: Transformer (deep learning architecture) Â§ History, and Seq2seq Â§ History
See also: Timeline of machine learning
Predecessors
For many years, sequence modelling and generation was done by using plain recurrent neural networks (RNNs). A well-cited early example was the Elman network (1990). In theory, the information from one token can propagate arbitrarily far down the sequence, but in practice the vanishing-gradient problem leaves the model's state at the end of a long sentence without precise, extractable information about preceding tokens.

A key breakthrough was LSTM (1995),[note 1] a RNN which used various innovations to overcome the vanishing gradient problem, allowing efficient learning of long-sequence modelling. One key innovation was the use of an attention mechanism which used neurons that multiply the outputs of other neurons, so-called multiplicative units.[13] Neural networks using multiplicative units were later called sigma-pi networks[14] or higher-order networks.[15] LSTM became the standard architecture for long sequence modelling until the 2017 publication of Transformers. However, LSTM still used sequential processing, like most other RNNs.[note 2] Specifically, RNNs operate one token at a time from first to last; they cannot operate in parallel over all tokens in a sequence.

Modern Transformers overcome this problem, but unlike RNNs, they require computation time that is quadratic in the size of the context window. The linearly scaling fast weight controller (1992) learns to compute a weight matrix for further processing depending on the input.[16] One of its two networks has "fast weights" or "dynamic links" (1981).[17][18][19] A slow neural network learns by gradient descent to generate keys and values for computing the weight changes of the fast neural network which computes answers to queries.[16] This was later shown to be equivalent to the unnormalized linear Transformer.[20][21]

Attention with seq2seq
Main article: Seq2seq Â§ History
The idea of encoder-decoder sequence transduction had been developed in the early 2010s; commonly cited as the originators that produced seq2seq are two concurrently published papers from 2014.[22][23]

A 380M-parameter model for machine translation uses two long short-term memories (LSTM).[23] Its architecture consists of two parts. The encoder is an LSTM that takes in a sequence of tokens and turns it into a vector. The decoder is another LSTM that converts the vector into a sequence of tokens. Similarly, another 130M-parameter model used gated recurrent units (GRU) instead of LSTM.[22] Later research showed that GRUs are neither better nor worse than LSTMs for seq2seq.[24][25]

These early seq2seq models had no attention mechanism, and the state vector is accessible only after the last word of the source text was processed. Although in theory such a vector retains the information about the whole original sentence, in practice the information is poorly preserved. This is because the input is processed sequentially by one recurrent network into a fixed-size output vector, which is then processed by another recurrent network into an output. If the input is long, then the output vector would not be able to contain all relevant information, degrading the output. As evidence, reversing the input sentence improved seq2seq translation.[26]

The RNNsearch model introduced an attention mechanism to seq2seq for machine translation to solve the bottleneck problem (of the fixed-size output vector), allowing the model to process long-distance dependencies more easily. The name is because it "emulates searching through a source sentence during decoding a translation".[4]

The relative performances were compared between global (that of RNNsearch) and local (sliding window) attention model architectures for machine translation, finding that mixed attention had higher quality than global attention, while local attention reduced translation time.[27]

In 2016, Google Translate was revamped to Google Neural Machine Translation, which replaced the previous model based on statistical machine translation. The new model was a seq2seq model where the encoder and the decoder were both 8 layers of bidirectional LSTM.[28] It took nine months to develop, and it outperformed the statistical approach, which took ten years to develop.[29]

Parallelizing attention
Main article: Attention (machine learning) Â§ History
Seq2seq models with attention (including self-attention) still suffered from the same issue with recurrent networks, which is that they are hard to parallelize, which prevented them from being accelerated on GPUs. In 2016, decomposable attention applied a self-attention mechanism to feedforward networks, which are easy to parallelize, and achieved SOTA result in textual entailment with an order of magnitude fewer parameters than LSTMs.[30] One of its authors, Jakob Uszkoreit, suspected that attention without recurrence would be sufficient for language translation, thus the title "attention is all you need".[31] That hypothesis was against conventional wisdom at the time, and even his father Hans Uszkoreit, a well-known computational linguist, was skeptical.[31] In the same year, self-attention (called intra-attention or intra-sentence attention) was proposed for LSTMs.[32]

In 2017, the original (100M-sized) encoder-decoder transformer model was proposed in the "Attention is all you need" paper. At the time, the focus of the research was on improving seq2seq for machine translation, by removing its recurrence to process all tokens in parallel, but preserving its dot-product attention mechanism to keep its text processing performance.[1] This led to the introduction of a multi-head attention model that was easier to parallelize due to the use of independent heads and the lack of recurrence. Its parallelizability was an important factor to its widespread use in large neural networks.[33]

AI boom era
Already in spring 2017, even before the "Attention is all you need" preprint was published, one of the co-authors applied the "decoder-only" variation of the architecture to generate fictitious Wikipedia articles.[34] Transformer architecture is now used alongside many generative models that contribute to the ongoing AI boom.

In language modelling, ELMo (2018) was a bi-directional LSTM that produces contextualized word embeddings, improving upon the line of research from bag of words and word2vec. It was followed by BERT (2018), an encoder-only Transformer model.[35] In 2019 October, Google started using BERT to process search queries.[36] In 2020, Google Translate replaced the previous RNN-encoderâ€“RNN-decoder model by a Transformer-encoderâ€“RNN-decoder model.[37]

Starting in 2018, the OpenAI GPT series of decoder-only Transformers became state of the art in natural language generation. In 2022, a chatbot based on GPT-3, ChatGPT, became unexpectedly[38] popular, triggering a boom around large language models.[39][40]

Since 2020, Transformers have been applied in modalities beyond text, including the vision transformer,[41] speech recognition,[42] robotics,[43] and multimodal.[44] The vision transformer, in turn, stimulated new developments in convolutional neural networks.[45] Image and video generators like DALL-E (2021), Stable Diffusion 3 (2024),[46] and Sora (2024), use Transformers to analyse input data (like text prompts) by breaking it down into "tokens" and then calculating the relevance between each token using self-attention, which helps the model understand the context and relationships within the data.

Training

While the primary focus of the paper at the time was to improve machine translation, the paper also discussed the use of the architecture on English Constituency Parsing, both with limited and large-sized datasets, achieving a high-score without specific tuning for the task indicating the promising nature of the model for use in a wide-variety of general purpose of seq2seq tasks.

Dataset

The English-to-German translation model was trained on the 2014 WMT (Workshop on Statistical Machine Translation) English-German dataset, consisting of nearly 4.5 million sentences derived from TED Talks and high-quality news articles. A separate translation model was trained on the much larger 2014 WMT English-French dataset, consisting of 36 million sentences. Both datasets were encoded with byte-pair encoding.

Hardware

The models were trained using 8 NVIDIA P100 GPUs. The base models were trained for 100,000 steps and the big models were trained for 300,000 steps - each step taking about 0.4 seconds to complete for the base models and 1.0 seconds for the big models. The base model trained for a total of 12 hours, and the big model trained for a total of 3.5 days. Both the base and big models outperforms the 2017 state-of-the-art in both English-German and English-French while achieving the comparatively lowest training cost.[1] The estimated computing cost was 0.089 petaFLOP-days.[47]

Hyperparameters and regularization

For their 100M-parameter Transformer model, the authors increased the learning rate linearly for the first 4000 (warmup) steps and decreased it proportionally to inverse square root of the current step number. Dropout layers were applied to the output of each sub-layer before normalization, the sums of the embeddings, and the positional encodings. The dropout rate was set to 0.1. Label smoothing was applied with a value of

Notes

Gated recurrent units (2014) further reduced its complexity.

Some architectures, such as RWKV or state space models, avoid the issue.