Review of Vaswani et al. (2017)

Matthew Evans

April 8, 2025

Overview

Previous methods[1] for machine translation processed text sequentially, yielding slow learning, especially for longer texts. Vaswani et al.[2] proposed a framework that focuses on key parts of the text simultaneously, enabling faster training and better translations through parallel processing. This approach simplified the process and enhanced performance while reducing architectural complexity.

Approach

Prior Work

Prior methods, such as the Extended Neural GPU[3], ByteNet[4], and ConvS2S[5], relied on convolutional architectures. While these models computed representations for all positions simultaneously, their ability to relate distant positions scaled poorly—linearly for ConvS2S and logarithmically for ByteNet. This inefficiency made capturing long-range dependencies challenging, requiring deeper networks and increasing computational costs.

End-to-end memory networks[6] explored the use of recurrent attention mechanisms rather than strict sequence-aligned recurrence, showing promising results on simpler language tasks such as question answering and language modeling. However, these approaches still inherited the limitations of recurrent processing, where the sequential nature of computation constrains parallelization and extends training times. This sequential bottleneck not only hinders the efficient learning of long-range relationships but also prevents models from fully utilizing the capabilities of modern GPU hardware for parallelization.

Novelty

The authors' *Transformer* model introduces a new architecture for sequence transduction tasks (e.g., translation) that entirely removes recurrence and convolution instead relying only on attention mechanisms, particularly *self-attention*, to model relationships between input and output tokens.

The proposed approach replaces sequence-aligned recurrent or convolutional networks with self-attention mechanisms, enabling simultaneous processing of all token positions via optimized matrix operations. This design significantly reduces training time and maximizes the use of efficient GPU operations.

The Transformer retains the encoder-decoder framework. Its encoder transforms the input sequence $x = (x_1, \dots, x_n)$ into continuous representations $z = (z_1, \dots, z_n)$ using stacked self-attention and feed-forward layers, with positional encodings adding order information. Its decoder generates the output sequence $y = (y_1, \dots, y_m)$ auto-regressively, considering both z and previously generated tokens. This design enhances translation quality and leverages parallel processing for efficiency in sequence transduction tasks.

Self-Attention

The transformer model uses self-attention, which enables every token to directly attend to every other token in the sequence, regardless of distance. Formally, given input embeddings $X \in \mathbb{R}^{n \times d_{\text{model}}}$, we compute:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$ are learned projection matrices.

The scaled dot-product attention[1] is then

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

This operation allows each token to attend to all others in the sequence via the dot products between query Q and key vectors K.

Multi-Head Attention

The authors extend the use of attention to so-called *multi-head attention*, enabling the model to capture multiple types of relationships in parallel by applying multiple attention functions simultaneously. For each *head* $i \in \{1, \ldots, h\}$:

$$Q_i = XW_i^Q, \quad K_i = XW_i^K, \quad V_i = XW_i^V$$

The attention for each head is

$$\text{head}_i = \text{Attention}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^\top}{\sqrt{d_k}}\right) V_i.$$

Finally, the multi-head attention is given as

$$\text{MultiHead}(X) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) W^O$$

where $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ is a learned output projection.

Positional Encoding

To inject information about token order into the model (which lacks recurrence or convolution), the authors add *positional encodings* to the input embeddings. These encodings use *sin* and *cos* functions of different frequencies:

$$\text{PE}_{(pos,\, 2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$
$$\text{PE}_{(pos,\, 2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

Where pos is the position (0-indexed) in the sequence, i is the dimension index, and d_{model} is the dimensionality of the model (e.g., 512). The positional encoding $\text{PE}(pos) \in \mathbb{R}^{d_{\text{model}}}$ is added element-wise to the input embedding $E(pos)$:

$$X_{pos} = E(pos) + \text{PE}(pos)$$

Strengths

The Transformer model offers several notable advantages over prior approaches.

- **Computational Efficiency**: Highly parallelizable architecture enables better performance at a fraction of the training cost.
- **Training Efficiency**: Trains 3 to 5 times faster compared to recurrent or convolutional models.
- **Interpretability**: Attention weights reveal which input tokens the model attends to, offering transparency into decision-making.
- **State-of-the-Art Performance**: Achieves higher BLEU scores than prior models, surpassing previous best results by over 2 BLEU on English-to-German translation.

[1] The authors use a form of attention known as multiplicative attention which lends itself well to the optimized hardware offered by GPUs. The authors scale the multiplicative attention by a factor of $\frac{1}{\sqrt{d_k}}$ in an effort to counteract the risk of vanishing gradients.

Considerations

The authors' approach, while groundbreaking, is not without its challenges.

- **Quadratic Complexity**: Self-attention scales as $\mathcal{O}(n^2)$ in time and memory, limiting efficiency on long sequences.
- **Lack of Sequential Inductive Bias**: Without recurrence or convolution, the model forfeits a useful sequential bias that is intrinsic to RNNs.
- **Auto-Regressive Inference Bottleneck**: Despite fast training, decoding remains sequential, slowing inference in generation tasks.
- **Limited Local Bias**: Unlike CNNs or RNNs, the model does not naturally emphasize local context, which may reduce efficiency in some domains.

Measures of Success

The authors validated their approach using standard translation quality metrics and training efficiency benchmarks. They measured translation performance primarily via the BLEU score, where their "big" model reached 28.4 on the WMT 2014 English-to-German task and 41.0 on the English-to-French task, surpassing previous state-of-the-art results by more than 2 BLEU points. Notably, the big model was trained in just 3.5 days using 8 P100 GPUs, while even their base model (trained in 12 hours) outperformed competitive systems at a fraction of the training cost (measured in FLOPs).

Impact

The Transformer model introduced by Vaswani et al. in *Attention Is All You Need* has had a profound and lasting impact on the field of machine learning, particularly natural language processing (NLP). Its attention-based architecture replaced recurrent and convolutional structures, enabling significantly greater parallelism and scalability. This innovation laid the foundation for nearly all modern large language models, including BERT[7], GPT[8], T5[9], and others, which dominate NLP benchmarks and real-world applications. The Transformer also catalyzed a paradigm shift toward pretraining on large corpora followed by task-specific fine-tuning. Beyond NLP, it has been successfully adapted to other modalities such as vision (Vision Transformers), audio, and multimodal systems. Moreover, its computational characteristics have driven extensive research into efficient attention mechanisms, model compression, and interpretability. The Transformer thus represents a unifying architecture with cross-domain applicability and has redefined the landscape of deep learning.

References

[1] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[3] Ł ukasz Kaiser and Samy Bengio. Can active memory replace attention? In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[4] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aäron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *CoRR*, abs/1610.10099, 2016.

[5] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122, 2017.

[6] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[8] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. Technical Report, OpenAI.

[9] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019.