TRANSFORMERS MODEL IN DEEP LEARNING

Deep Learning and Computer Vision, 2024

No Need for LSTM or RNN, instead “attention is all we need.”

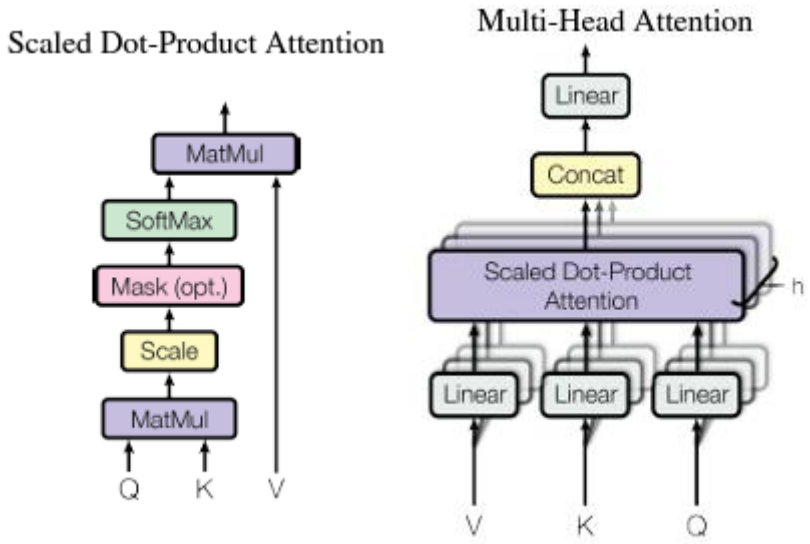
Abukar Ali

Abstract

The attention mechanism according to the publications by Vaswani et al (2017) is described as a mapping a query and a set of keys-value pairs to an output. The query, keys and values outputs are all vectors. The output is then computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. This single head-self attention is referred to as "scaled Dot-Product Attention". The dot product of the query with all the keys are computed. The results are scaled by sqrt (dk). The scaled results are then passed to a SoftMax function to obtain the weights on the values.

These computations using the attention function are represented in following compact form.

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



However, the Multi-head Attention runs through the attention mechanism several times in parallel. This means that the attention module repeats its computation multiple times in parallel. Each of these is called an attention head. The attention head module splits its query, key and value parameters N-ways and passes each split independently through a separate head. All these similar attention calculations are then combined together to produce a final attention score. This is called the multi-head attention

Vaswani et al. (2017) describe the multi-head technically as a applying the single head self-attention h-times by linearly projecting the queries, keys and values h-times with different learned linear projections to d(q)[1], d(k), d(v) dimensions. Each of the projected version of the queries, keys and values are then used to perform the attention function in parallel. Unlike single head attention, the multi-head attention allows the models to jointly attend information from different representation subspaces at different positions. Vaswani et al (2017).

1.1 INTRODUCTION

The Transformer model (which is also known as Vanilla Transformer) originally proposed by Vaswani et al. (2017) is described as neural network architecture which is mainly based on attention mechanism. The authors presented innovative architecture for Natural Language processing that replaced traditional Recurrent Natural Network (RNN) with self-attention model[2]. The self-attention framework enabled the model to focus on different parts of the input sequence when generating the output, leading to significant improvements in performance.

2.1 ANALYSIS

The self-attention mechanism within the Transformer operates by calculating the weighted sum of an input sequence. The weights are determined on the basis of the similarities between each element in the input vector and a query vector. The query vector is derived from the hidden state of the decoder, enabling the model to focus (attend) to different parts of the input sequence depending on the context.

The key advantage of the transformer model is its ability to capture long-range dependencies in a sequence, which is a significant challenge for RNN-based architectures. The self-attention mechanism allows the transformer to model complex relationships between different parts of the input sequence, leading to more accurate and coherent predictions. Since its introduction, the transformer has undergone several modifications

[1] There may be a typo in the original paper
[2] https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

and extensions. Variants such as the BERT and the GPT series have further improved upon the original architecture and have demonstrated impressive performance on a wide range of natural language processing tasks (Maarseveen, Henri).

Amarian et al. (2023) highlighted the advantages of attention layers of recurrent and convolutional network. The two most important being their lower computational complexity and their higher connectivity, especially useful for learning long-term dependencies in sequence. The authors provide a catalog of extended versions of the transformer since the introduction of the vanilla transformer model. These models include the aforementioned BERT variant - Bidirectional Encoder Representations from Transformers (BERT), Generative Pre-trained Transformer (GPT) originated by Open AI, Attention heads and multi-head attention, Sparce Attention, Learned positional Embeddings[3]. The BERT model is able to capture both the meaning and the order in a sentence, and is able to handle complex and ambiguous language such as sarcasm, negation etc. There are still further contributions in this field, however, these models are mainly categorized into three branches: the Encoder, Decoder and Encoder-Decoder branches (figure 1).

3.1 CHALLENGES AND FUTURE DIRECTION

Lin et al., (2022) conducted a survey and provided a comprehensive overview of the transformer model and its variations. They concluded that most Most of the existing works improve Transformer from different perspectives, such as efficiency, generalization, and applications. The improvements include incorporating structural prior, designing lightweight architecture, pre-training, etc.

The survey also provides an overview of the current challenges despite the innovations adopted from transformer models. In addition to recent issues that are related to efficiency and generalization, the key suggested improvements include:

[3] https://huggingface.co/docs/transformers/model_doc/bert

The need for further analysis to gain further theoretical understanding of the transformer's capabilities. The need for further improvements to model global interactions and the use of alternative neural networks, for example, memory enhanced models. Further improvements are recommended in the design of the intra-model and cross-modal attention.

S. Islam et al. (2024) have conducted one of the most comprehensive surveys to date on Transformer models and their applications. They report that computer vision and NLP collectively constitute just over 70% of Transformer applications, whereas applications in Audio and Speech currently account for only about 11%. This suggests a potential area of focus in the coming years.

Khoei. T., et al. (2023) provided a comprehensive overview of deep learning models and highlighted a number of challenges as well as future direction. One of the key challenges involves the availability and quality of data. Although data augmentation is utilized usually, this approach may not be sufficient to generate enough training data to satisfy the requirements of Deep Learning models which could lead to overfitting issues. Another key challenge in developing and deploying deep learning models is related to the need to address biases, discriminations and other social implications embedded within all Artificial Intelligence based models. Interpretability and explainability of deep learning models. These models may be thought of as "black box" models. The absence of transparency poses a barrier to trust and adoption of these models, particularly in critical applications such as finance and healthcare.

Other areas that remain challenging include "catastrophic forgetting" where a model forgets previously learned information which could lead to degradation in performance on tasks that were previously well learned. This issue arises in models that utilize a large number of parameters such as transformer-based models. "Save learning" – considers the safety and risk associated with Deep learning models. For example, ground or aerial robots can result in undesirable outcomes such as casualties. It is crucial to ensure AI models incorporate all safety related properties, risk estimation, dealing with uncertainty in the data, and detecting abnormal system behaviours and unforeseen events to ensure

safety and avoid catastrophic failures and hazards. The research in this area is still at a very early stage. Overcoming these challenges perfectly align with the potential of combining transformers with other deep learning models[4] such as graph neural networks and reinforcement learning algorithms, to create more sophisticated models that can better handle complex tasks while addressing all ethical and safety issues.

[4] See figure 2: schematic review of the models in deep Learning. Khoei. T., et al. (2023)

Appendix

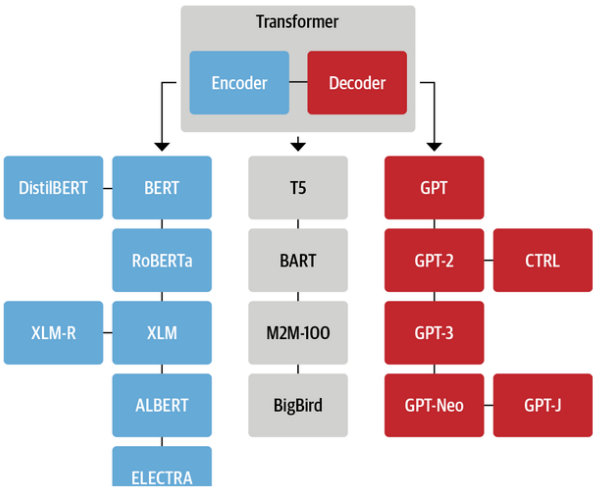


Figure 1: "Transformer Tree of Life": an overview of the most prominent transformer architectures.

Source: Tunstal et al., Natural Language Processing with Transformers, Revised edition. O'Reilly Media Publications, 2022.

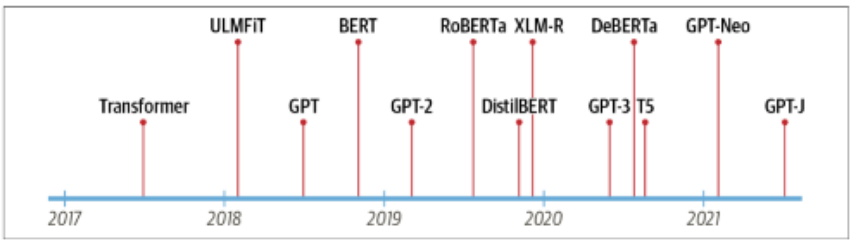
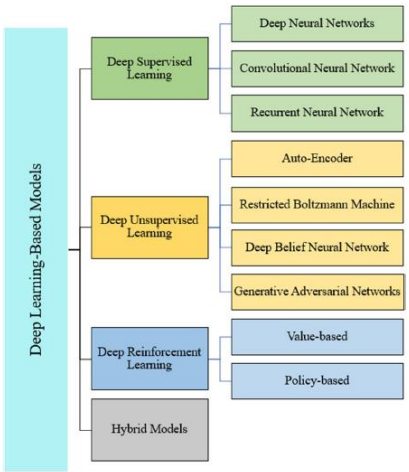


Figure 1-1. The transformers timeline

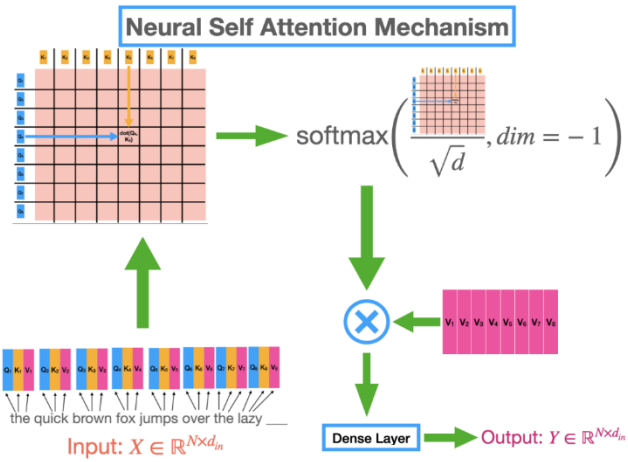
Source: Tunstal et al., Natural Language Processing with Transformers, Revised edition. O'Reilly Media Publications, 2022.

Figure 2: Schematic view of the models in Deep Learning: Khoei. T., et al. (2023)



A good overview of the self-attention mechasim calculations:

https://learnopencv.com/attention-mechanism-in-transformer-neural-networks/



References

Talaei Khoei, T., Ould Slimane, H. & Kaabouch, N. Deep learning: systematic review, models, challenges, and research directions. *Neural Comput & Applic* **35**, 23103–23124 (2023). https://doi.org/10.1007/s00521-023-08957-4

S. Islam, Elmekki, H., Elsebai. A., et al., (2024). "A comprehensive survey on applications of transformers for deep learning tasks. Expert Systems with Applications | An international Journal.

Turner, Richard E. "An introduction to Transformers". Department of Engineering, University of Cambridge, UK. Microsoft Research, Cambridge, UK.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need.

Amatriain, Xavier., Shankar, Ananth., Bing, et al. (2023). "Transformer models: an introduction and catalog".

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, *3*, 111–132.

Maarseveen, Henri., Attention is All You Need: The Game-Changing Paper That Transformed NLP

Important Website/links/channels, etc

Python Projects with Source Code | Aman Kharwal (thecleverprogrammer.com)
LeNet-5 Architecture using Python | Aman Kharwal (thecleverprogrammer.com)
https://magazine.sebastianraschka.com/
https://magazine.sebastianraschka.com/p/understanding-and-coding-self-attention
https://d2l.ai/chapter_attention-mechanisms-and-transformers/multihead-attention.html
https://github.com/rasbt/machine-learning-book
https://www.youtube.com/watch?v=0PjHri8tc1c&list=PLTKMiZHVd_2KJtIXOW0zFhFfBaJJilH51&index=165&ab_chan
Attention is all you need (Transformer) - Model explanation (including math), Inference and Training (youtube.com)
Illustrated Guide to Transformers- Step by Step Explanation | by Michael Phi | Towards Data Science
https://maximliu-85602.medium.com/learn-cnn-and-pytorch-through-understanding-torch-nn-conv2d-class-54ad9
https://www.saedsayad.com/clustering_kmeans.htm
The Illustrated Word2vec – Jay Alammar – Visualizing machine learning one concept at a time. (jalammar.github.io)
Transformers — Visual Guide (mayurji.github.io)
Illustrated Guide to Transformers- Step by Step Explanation | by Michael Phi | Towards Data Science
Jay Alammar – Visualizing machine learning one concept at a time. (jalammar.github.io)

Amazing Website. All the Generative AI models (code). This is gold
https://nn.labml.ai/

https://towardsdatascience.com/transformers-explained-visually-part-3-multi-head-attention-deep-dive-1c1ff1024

Diffusion Models

https://kailashahirwar.medium.com/a-very-short-introduction-to-diffusion-models-a84235e4e9ae
https://towardsdatascience.com/diffusion-models-made-easy-8414298ce4da
https://theaisummer.com/diffusion-models/
https://theaisummer.com/topics/computer-vision/

https://erdem.pl/2023/11/step-by-step-visual-introduction-to-diffusion-models#forward-diffusion-diagram

https://medium.com/@kemalpiro/step-by-step-visual-introduction-to-diffusion-models-235942d2f15c
https://medium.com/@kemalpiro/step-by-step-visual-introduction-to-diffusion-models-235942d2f15c
https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
https://keras.io/examples/generative/ddim/
https://learn.deeplearning.ai/diffusion-models/lesson/1/introduction
https://www.youtube.com/watch?v=a4Yfz2FxXiY&ab_channel=DeepFindr
https://github.com/diff-usion/Awesome-Diffusion-Models
https://lilianweng.github.io/posts/2021-07-11-diffusion-models/#nice
https://maciejdomagala.github.io/generative_models/2022/06/06/The-recent-rise-of-diffusion-based-models.html#
https://www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction/
https://theaisummer.com/diffusion-models/?fbclid=IwAR1BIeNHqa3NtC8SL0sKXHATHkUYphNH-8IGNoO3xZhSKM

https://rpubs.com/eR_ic/ddpms

https://betterprogramming.pub/diffusion-models-ddpms-ddims-and-classifier-free-guidance-e07b297b2869
https://learnopencv.com/denoising-diffusion-probabilistic-models/#gaussian-distribution

https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/annotated_diffusion.ipynb#

https://medium.com/@kemalpiro/step-by-step-visual-introduction-to-diffusion-models-235942d2f15c
https://erdem.pl/2023/11/step-by-step-visual-introduction-to-diffusion-models

https://learn.deeplearning.ai/diffusion-models/lesson/3/sampling

https://ayandas.me/blog-tut/2021/12/04/diffusion-prob-models.html

https://huggingface.co/blog/annotated-diffusion
...also access the colab directly here

Best Books: Diffusion
https://learning.oreilly.com/library/view/generative-deep-learning/9781098134174/ch08.html#id120
https://learning.oreilly.com/library/view/hands-on-generative-ai/9781098149239/ch02.html#id71

https://github.com/CroitoruAlin/Diffusion-Models-in-Vision-A-Survey?tab=readme-ov-file#1

https://magic-with-latents.github.io/latent/posts/ddpms/part3/

https://pub.towardsai.net/diffusion-models-vs-gans-vs-vaes-comparison-of-deep-generative-models-67ab93e0d9a

Gold
https://learnopencv.com/denoising-diffusion-probabilistic-models/
https://learnopencv.com/image-generation-using-diffusion-models/
https://yang-song.net/blog/2021/score/

Youtube: KL divergence, Evidence Lower Bounnd, Max Likelyhood tutorials
https://www.youtube.com/watch?v=IXsA5Rpp25w&ab_channel=KapilSachdeva
https://www.youtube.com/watch?v=NyH9K3stvP8&ab_channel=KapilSachdeva
https://www.youtube.com/watch?v=q0AkK8aYbLY&ab_channel=ritvikmath

CNN Related Material (old to new)

https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/