TECHNISCHE UNIVERSITÄT
ILMENAU

Fakultät für Informatik und Automatisierung

Data-intensive Systems and Visualization Group

Attention is all you need! But which one?

Research Project

Taimur Ibrahim

Betreuer: M. Sc. Dominik Walther

Verantwortlicher Hochschullehrer: Prof. Dr.-Ing. Patrick Mäder

Das Masterarbeit wurde am Datum bei der Fakultät für Informatik und Automatisierung der Technischen Universität Ilmenau eingereicht.

Abstract

The success of the transformer architecture in the NLP domain can mainly be credited to its innovative self-attention mechanism which allows it to capture and model the relations between text sequences accurately. However, the same self-attention mechanism ends up holding back the architecture when applied to time-series modeling. The two main factors that make self-attention unsuitable for time-series modeling are its inability to prioritize specific tokens in the sequence and its quadratic time complexity which gets prohibitive for very long sequences. This paper will explore alternative attention mechanisms that try to overcome these limitations like Latent Attention, Sparse Attention, Informer's Attention, and Longformer's Attention, and evaluate their effect on predictive performance while also measuring computational efficiency and other potential trade-offs.

Der Erfolg der Transformer-Architektur im NLP-Bereich ist vor allem ihrem innovativen Mechanismus der Selbstaufmerksamkeit zu verdanken, der es ihr ermöglicht, die Beziehungen zwischen Textsequenzen genau zu erfassen und zu modellieren. Derselbe Mechanismus der Selbstbeobachtung ist jedoch ein Hindernis für die Architektur, wenn sie auf die Modellierung von Zeitreihen angewendet wird. Die beiden Hauptfaktoren, die die Selbstaufmerksamkeit für die Modellierung von Zeitreihen ungeeignet machen, sind ihre Unfähigkeit, bestimmte Token in der Sequenz zu priorisieren, und ihre quadratische Zeitkomplexität, die bei sehr langen Sequenzen untragbar wird. In diesem Beitrag werden alternative Aufmerksamkeitsmechanismen untersucht, die versuchen, diese Einschränkungen zu überwinden, wie z. B. Latent Attention, Sparse Attention, Informer's Attention und Longformer's Attention, und ihre Auswirkungen auf die Vorhersageleistung bewertet, wobei auch die Recheneffizienz und andere potenzielle Kompromisse gemessen werden.

Selbstständigkeitserklärung: „Hiermit versichere ich, dass ich diese Masterarbeit selbststständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe. Alle von mir aus anderen Veröffentlichungen übernommenen Passagen sind als solche gekennzeichnet. “

Ilmenau, Datum

Taimur Ibrahim

Contents

Introduction 1

In 2017, the Attention Is All You Need [Vaswani et al., 2017] paper was published. The transformer architecture described in the paper was designed for machine translation tasks in particular. However, researchers soon found that it was applicable to a much wider array of Natural Language Processing (NLP) tasks and gave state-of-the-art results in them. This realization led to great advances in the field of NLP, and variants of the same architecture continue to push the boundaries of what is possible. The success of the transformer architecture in the NLP domain understandably sparked interest in exploring its viability in other domains, including time-series modeling.

Accurate modeling of time-series data is crucial for decision-making and planning in domains such as finance, healthcare, and energy management. Traditional methods like ARIMA and exponential smoothing, as well as machine learning approaches such as recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), have been used extensively for this purpose. However, these methods often have trouble capturing the long-range dependencies and complex temporal patterns found in time-series data.

Transformers, due to their self-attention mechanism, have proven to be quite capable of learning long-range dependencies in textual data. However, two major problems arise when we try to apply this same mechanism to time-series data. Firstly, the self-attention mechanism assigns equal importance to all values in the input sequence whereas recent values or local patterns are often more important. Secondly, self-attention has a quadratic time complexity which makes it prohibitively expensive to compute on long input sequences.

This paper will explore different attention mechanisms that can replace the default implementation in the Transformer architecture to potentially enhance its applicability and performance in time-series modeling. By comparing the standard self-attention mechanism with approaches such as Latent Attention, Sparse Attention, Informer's Attention, and Longformer's Attention, the goal of this project is to determine which mechanism best captures the temporal dependencies in time-series data while balancing predictive performance and computational efficiency.

Background

2

2.1 Transformers

2.1.1 Architecture

As shown in **Figure 2.1**, the transformer model consists of two main blocks. The left and right blocks are called the **encoder** and **decoder** respectively. This specific architecture was needed because the original model was designed for the task of machine translation. The encoder generated a vector representation for the input sequence in the source language and the decoder used this vector to generate the corresponding vector in the target language.

2.1.2 Attention

Self Attention

At its core, the Transformer model builds on the concept of self-attention, a mechanism that computes the representation of a sequence by relating each element to every other element in the same sequence.

In [Vaswani et al., 2017], the authors named their attention implementation **Scaled Dot-Product Attention**. It is calculated via the following equation:

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

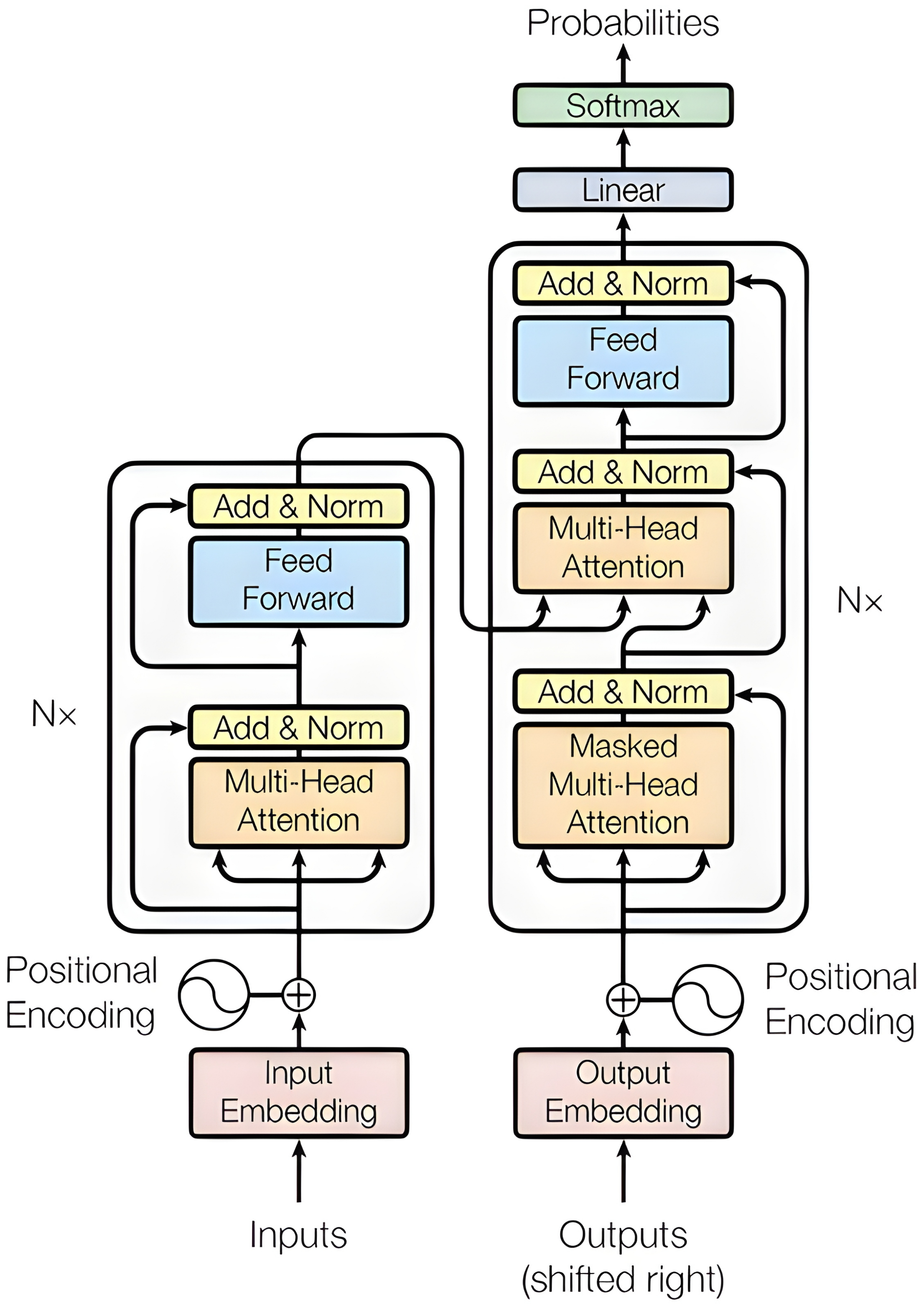

Figure 2.1: Original Transformer model architecture. Image taken from [Vaswani et al., 2017].

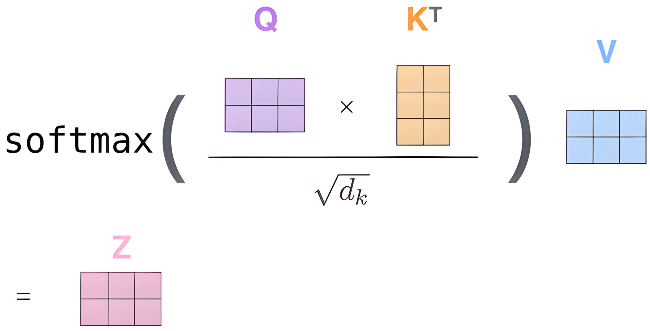

Figure 2.2: Visual representation of the matrices involved in the self-attention calculation. Image taken from [Alammar, 2018].

As shown in 2.2, the attention calculation makes good use of matrix multiplications but that also leads to it scaling quadratically in terms of both memory and time complexity.

Multi-Head Attention

In the paper, the authors found that instead of using just one attention function, their model produced better results when they introduced a mechanism called **Multi-head attention**. This mechanism improved the capabilities of the attention layer by allowing the model to focus on different parts of the input sequence.

The actual calculation is straightforward to understand. Instead of one attention function, we now have multiple "heads" which are essentially attention functions with their own randomly initialized weight matrix. As shown in 2.3, the number of attention heads in the layer is defined as **h**. In [Vaswani et al., 2017] the authors used 8 heads for their attention layer. Each of these heads pro-

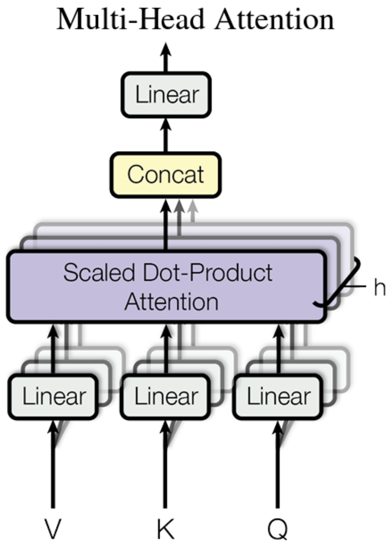

Figure 2.3: Multi-head attention visualized. Image taken from [Vaswani et al., 2017].

duces an output vector Z1, Z2, Z3 .... Zh using the same calculations as described in 2.2. All of these output vectors are then concatenated and multiplied by a new weight matrix W0 which gives us our final output vector Z.

In [Vaswani et al., 2017] the calculations for the multi-head attention mechanism are described by the following equations:

$$\begin{gathered}\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O \\ \text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)\end{gathered}$$

Here, Q, K, and V are the query, key, and value matrices, respectively, and W_i^Q, W_i^K, and W_i^V are the learned projection matrices for the i-th head. W^O is the output projection matrix.

2.2 Alternative Attention Mechanisms

Given the constraints of the standard attention mechanism, exploring alternative attention mechanisms that can reduce computational complexity and better capture local and temporal relationships within data is crucial.

Several studies have attempted to adapt the Transformer architecture for time-series analysis. For instance, the Informer model [Zhou et al., 2021b] introduces a ProbSparse self-attention mechanism to reduce the quadratic computational complexity of standard attention, making it more feasible for long time-series sequences. Similarly, the Longformer model [Beltagy et al., 2020] uses a combination of local and global attention to handle long documents, which could apply to long time-series sequences as well.

Other approaches include the use of Sparse Attention [Child et al., 2019], which selectively attends to a subset of relevant tokens, reducing computational overhead, and Latent Attention [Dolga et al., 2024], which uses latent variables to capture underlying patterns in the data. These alternative attention mechanisms offer potential solutions to the limitations of standard Transformers in time-series applications.

Related Work 3

This section gives an overview of mechanisms in current literature that could prove beneficial for adapting Transformers for time-series modeling.

3.1 Sparse Attention

Sparse Attention mechanisms, such as those proposed by Child et al. [Child et al., 2019], reduce the computational complexity of the standard self-attention by restricting the attention scope to a sparse set of tokens. This approach makes use of sparsity patterns to focus computation on the most relevant parts of the sequence, making it scalable to longer time-series.

3.2 Latent Attention

Latent Attention [Dolga et al., 2024] introduces latent variables to capture the structure of the input data. This method improves the model's ability to identify hidden patterns and dependencies that are not immediately apparent from the raw data. By incorporating latent variables, Latent Attention can improve the representation of complex temporal relationships.

3.3 Informer

The Informer model [Zhou et al., 2021b] addresses the quadratic complexity of the standard attention mechanism by introducing ProbSparse self-attention. This approach selectively focuses on a subset of relevant tokens, significantly reducing computational overhead. Informer also utilizes a distilling operation to handle the redundancy in time-series data, effectively compressing long sequences while retaining essential information. This makes Informer particularly suitable for long-term time-series forecasting.

3.4 Longformer

The Longformer model [Beltagy et al., 2020] introduces a combination of local and global attention mechanisms to manage long documents and, by extension, long time-series data. The model employs a sliding window approach for local attention, ensuring that each token attends to its neighbors, while global attention is applied to a selected subset of tokens to capture broader context. This dual attention mechanism balances the need for detailed local information and overall sequence comprehension, making it efficient for long sequences.

3.5 Other Notable Approaches

Several other models and methods have also contributed to the advancement of Transformers in time-series analysis. For example, the LogSparse Transformer [Li et al., 2020] adapts the attention mechanism to focus logarithmically more on recent tokens than distant ones, aligning better with the typical relevance patterns in time-series data. Additionally, the Reformer [Kitaev et al., 2020] employs locality-sensitive hashing to reduce the complexity of self-attention, making it feasible to process very long sequences efficiently.

Methods

4

4.1 Data

For the purpose of this paper, the FordA dataset will be used to train and evaluate the different models. This data was originally used in a competition in the 2008 IEEE World Congress on Computational Intelligence. The classification problem is to diagnose whether a certain symptom exists or not in an automotive subsystem. Each case consists of 500 measurements of engine noise.

Train size	3601
Test size	1320
Missing values	No
Number of classes	2
Time series length	500

Table 4.1: Summary of Dataset Properties

4.2 Model Architecture

For time-series classification, an encoder-only transformer is a suitable choice. The model will be implemented in a modular fashion so that the attention block can be easily swapped out for other implementations.

4.3 Attention Mechanisms

4.3.1 Scaled Dot-product Attention

This is the default implementation of the attention mechanism that will act as the baseline for all future comparisons.

4.3.2 Sparse Attention

...

4.4 Model Training

The models were trained in a cloud-based jupyter environment on a system with [numberofgpus] [gpuname] GPU and [systemram] RAM.

4.5 Evaluation Metrics

The different models will be judged based on various metrics such as training time, prediction accuracy(RMSE, MAE), inference time, model size and memory footprint

4.6 Reproducibility

The data and code for this paper is available on GitLab(https://gitlab.tu-ilmenau.de/dowa4213/research-project-attention) so that any interested parties can reproduce the results or extend the work. Access will be granted on a case-by-case basis.

Experiments 5

describe experimental setup,

Summary 6

Appendix

A

List of Figures

List of Tables

Bibliography

[Alammar, 2018] Alammar, J. (2018). The illustrated transformer.

[Beltagy et al., 2020] Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

[Child et al., 2019] Child, R., Gray, S., Radford, A., and Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

[Dolga et al., 2024] Dolga, R., Cobzarenco, M., and Barber, D. (2024). Latent attention for linear time transformers. *arXiv preprint arXiv:2402.17512*.

[Kitaev et al., 2020] Kitaev, N., Łukasz Kaiser, and Levskaya, A. (2020). Reformer: The efficient transformer.

[Li et al., 2020] Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., and Yan, X. (2020). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Wen et al., 2022] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L. (2022). Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.

[Zhang et al., 2021] Zhang, B., Titov, I., and Sennrich, R. (2021). Sparse attention with linear units. *arXiv preprint arXiv:2104.07012*.

[Zhou et al., 2021a] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. (2021a). Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115.

[Zhou et al., 2021b] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. (2021b). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115.