Transformer gibt es nicht nur im Kino

Jonathan Arns
Hochschule Mannheim
Fakultät für Informatik
Paul-Wittsack-Str. 10
68163 Mannheim
jonathan.arns@stud.hs-mannheim.de

***Abstract*—abstract**

I. Einleitung

Natural Language Processing (NLP), die Verarbeitung Menschlicher Sprache, ist ein Anwendungbereich für Machine Learning, der bereits erheblich von deep neural networks (DNN) profitiert hat. Die zwei dominanten Arten von DNN Architekturen dabei waren lange Zeit recurrent neural networks (RNN) und convolutional neural networks (CNN). [1]

2017 stellten Vaswani et al. [2] mit dem Transformer eine neue DNN Architektur vor, die seitdem unter Anderem große Aufmerksamkeit durch die erfolgreiche Verwendung in OpenAIs GPT-2 und GPT-3 Modellen erlangte.

II. Sequence to Sequence Learning

Traditionell waren DNNs trotz ihrer hohen Flexibilität und Effektivität für viele Aufgaben beschränkt auf Probleme, deren Eingaben und Ausgaben sich sinvoll in Vektoren mit fester Länge codieren lassen, da neuronale Netze generell eine feste Anzahl an Eingabe- und Ausgabeneuronen haben. Das ist zwar für viele Klassifizierungsprobleme und in der Bildverarbeitung kein Problem, sehr wohl aber für beispielsweise NLP, da die Länge von Texten im Vorfeld nicht immer bekannt ist. [3]

Dieses Problem wird mittels Sequence to Sequence (Seq2Seq) Learning gelöst, mit dessen Hilfe beliebig lange Sequenzen von Elementen als Eingabe in vollkommen andere Sequenzen, anderer Länge und aus anderen Elementen, transformiert werden können. [3]

Die meisten Seq2Seq Modelle bestehen aus einem Encoder und einem Decoder [2]. Der Encoder codiert die Eingabe Sequenz in einer höher-dimensionale Vektorrepräsentation, die dann vom Decoder in eine Ausgabe Sequenz umgewandelt wird. Die Encoder und Decoder selbst sind in der Regel RNNs, beispielsweise mit der LSTM Architektur.

III. Modell Architektur

IV. Attention

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_K}})V \quad (1)$$

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \quad (2)$$

$$\text{mit } head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

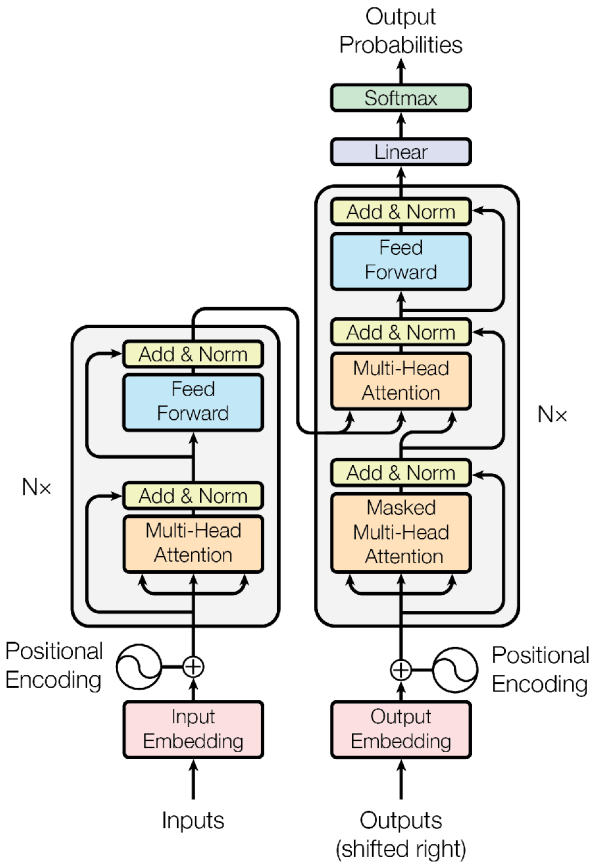

Fig. 1. Der Encoder (links) besteht aus einer Reihe von N identischen Modulen mit jeweils zwei Untermodulen.
Der Decoder (rechts) besteht auch aus einer Reihe von N identischen Modulen, jedoch mit jeweils drei Untermodulen. Die erste Schicht ist eine Multi-Head Attention Layer [2]

V. Training

VI. Fazit

Das ist ein Fazit

References

[1] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. “Comparative Study of CNN and RNN for Natural Language Processing”. In: *arXiv e-prints* (2017). arXiv: 1702.01923.

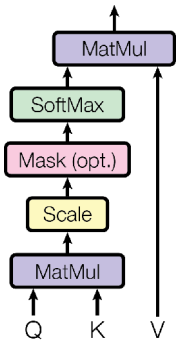

Fig. 2. Scaled dotproduct attention [2]

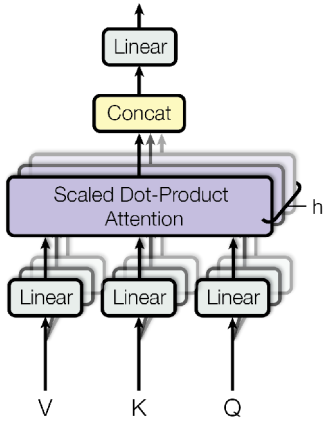

Fig. 3. Multi-Head Attention [2]

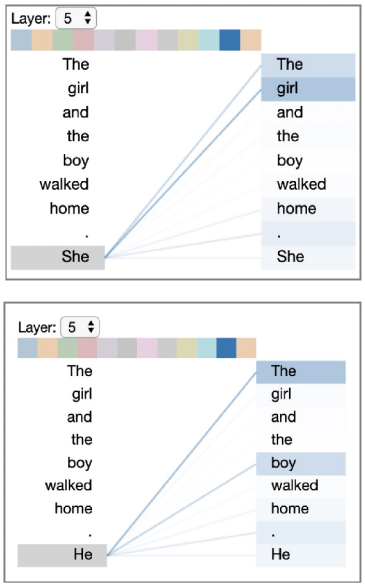

Fig. 4. Visualization of attention in GTP-2 [4]

[2] Ashish Vaswani et al. "Attention is All You Need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 9781510860964.

[3] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to Sequence Learning with Neural Networks". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14. Montreal, Canada: MIT Press, 2014, pp. 3104–3112.

[4] Jesse Vig. "A Multiscale Visualization of Attention in the Transformer Model". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 37–42. DOI: 10.18653/v1/P19-3007. URL: https://www.aclweb.org/anthology/P19-3007.