Transformer

Autor und Herausgeber

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland (ORCID 0000-0001-7538-0990)
Editors: Konstantina Olioumtsevits (ORCID 0000-0003-2697-8659), Ekkehard Schnoor (ORCID 0000-0001-8252-1499), Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Zusammenfassung

Ein Transformer ist ein künstliches neuronales Netz (KNN), das durch die Verkettung von Schichten aufgebaut wird, von denen jede die Merkmale eines Datenpunkts transformiert, der aus kleineren Einheiten, sogenannten Tokens, besteht, wie etwa den Wörtern eines Textes oder den Bildausschnitten eines Bildes. Die Merkmalsvektoren der Tokens werden zu einer Matrix gestapelt; jede Schicht bildet diese Matrix auf eine derselben Form ab, sodass Schichten frei zu einem tiefen KNN verkettet werden können. Die Schichten wechseln sich ab zwischen Token-Mischung, die Informationen über Tokens hinweg kombiniert, und einem positionsweisen mehrschichtiges Perzeptron (MLP), das jedes Token separat transformiert.

Definition

Ein Text besteht aus Wörtern, und ein Bild besteht aus Pixel-Ausschnitten (siehe Abb. 1): Die Datenpunkte vieler maschinelles Lernen (ML)-Anwendungen bestehen aus kleineren Einheiten, sogenannten Tokens. Ein Transformer ist ein künstliches neuronales Netz (KNN), das durch die Verkettung von Schichten aufgebaut ist, von denen jede die Merkmale eines solchen Token-basierten Datenpunkts transformiert (Vaswani et al., 2017).

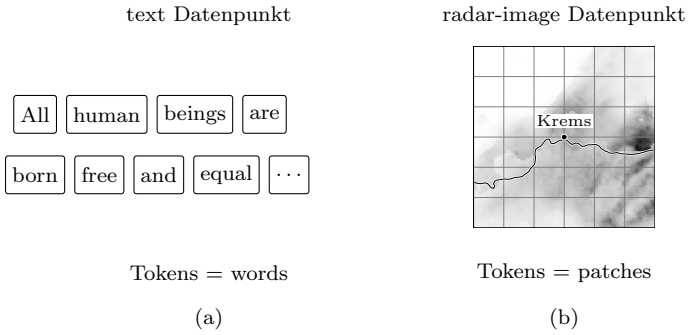


Abbildung 1: Zwei Datenpunkte, die aus Tokens bestehen. (a) Die einleitenden Worte der Allgemeinen Erklärung der Menschenrechte (United Nations General Assembly, 1948); jedes Wort ist ein Token. (b) Ein wetterradarbasiertes Bild des einstündigen Niederschlags rund um Krems an der Donau, Österreich (20. August 2026; INCA-Analyse (Integrated Nowcasting through Comprehensive Analysis), GeoSphere Austria), mit 96×96 Pixeln zu je 1 km; dunklere Pixel zeigen mehr Niederschlag an, und dünne graue Linien unterteilen das Bild in 16×16-Pixel-Ausschnitte, von denen jeder ein Token ist. Die schwarze Linie, die mit einer weißen Umrandung gezeichnet ist, damit sie über dunklen Pixeln sichtbar bleibt, markiert den Verlauf der Donau, und der Punkt in der Bildmitte die Lage von Krems (Flussverlauf: OpenStreetMap-Mitwirkende). Daten generiert durch `pythondemos/transformer.py`

Die Merkmalsvektoren (oder Einbettungen) $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)} \in \mathbb{R}^d$ der n Tokens werden zu einer Matrix $\mathbf{X} = \left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\right)^\top \in \mathbb{R}^{n \times d}$ gestapelt. Jede Schicht des ursprünglichen Transformers bildet eine solche Matrix auf eine transformierte Matrix derselben Form ab, sodass Schichten frei zu einem tiefen KNN verkettet werden können (Vaswani et al., 2017, Abschn. 3.1). Ein Transformer, der einen Satz übersetzt, transformiert beispielsweise die gestapelten Worteinbettungen Schicht für Schicht in Merkmalsvektoren, aus denen die übersetzten Wörter vorhergesagt werden (Vaswani et al., 2017); Transformer verarbeiten auch Bilder, indem sie Bildausschnitte als Tokens verwenden (Dosovitskiy et al., 2021). Moderne große Sprachmodelle (LLM) wie GPT-3 sind Transformer, die darauf trainiert sind, das nächste Token eines Textes vorherzusagen (Brown et al., 2020).

Die Schichten eines Transformers wechseln sich zwischen zwei Arten von Transformationen von $\mathbf{X}$ ab (siehe Abb. 2). Eine Token-mischende Schicht bildet $\mathbf{X}$ auf eine neue Matrix $\widetilde{\mathbf{X}}$ ab, deren jede Zeile eine gewichtete Summe linearer Bilder der Zeilen von $\mathbf{X}$ ist, mit Gewichten, die von $\mathbf{X}$ selbst abhängen können; die Tokens interagieren dadurch miteinander. Im ursprünglichen Transformer wird die Token-Mischung durch Multi-Head-Attention implementiert, deren Ausgabe für jedes Token eine gewichtete Summe der Wertvektoren ist, also der Zeilen von $\mathbf{X}$ multipliziert mit einer gelernten Matrix (Vaswani et al., 2017, Abschn. 3.2). Attention unterscheidet Transformer von rekurrente neuronale

Netze (RNN), welche die Tokens eines Datenpunkts sequenziell statt parallel verarbeiten (Vaswani et al., 2017, Abschn. 1). Ein positionsweises mehrschichtiges Perzeptron (MLP) transformiert anschließend jede Zeile von $\widetilde{\mathbf{X}}$ separat (Vaswani et al., 2017, Abschn. 3.3). Bei jeder der beiden Transformationen addiert eine Residualverbindung die Eingabe der Transformation zu ihrer Ausgabe, und eine anschließende Schichtnormalisierung standardisiert jede Zeile der resultierenden Matrix (Vaswani et al., 2017, Abschn. 3.1).

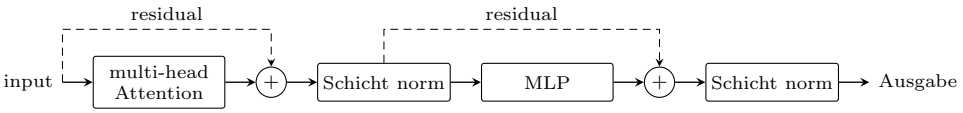


Abbildung 2: Ein Transformer-Block. Jede der beiden Transformationen, Multi-Head-Attention zur Token-Mischung und ein positionsweises MLP, wird durch eine Residualverbindung (gestrichelter Pfeil) überbrückt, deren Ausgabe wieder hinzuaddiert und durch eine Schichtnormalisierung geleitet wird. Das größte Llama-3-LLM stapelt mehr als 120 solcher Blöcke (Grattafiori et al., 2024, Abschn. 3.2)

Die von aufeinanderfolgenden Schichten erzeugten Matrizen repräsentieren die Tokens auf zunehmend abstrakten Ebenen. Dieser Fortschritt wird durch Probing (Sondierung) gemessen: Ein einfacher Klassifikator wird darauf trainiert, eine gewählte Eigenschaft aus der Ausgabe einer einzelnen Schicht vorherzusagen, und die Tiefe, ab der die Eigenschaft vorhersagbar wird, lokalisiert sie im Netzwerk. In einem Sprach-Transformer sind Eigenschaften auf Wortebene aus der Ausgabe früher Schichten vorhersagbar. Ein Beispiel ist eine Wortart-Markierung (Part-of-Speech-Tag), die angibt, ob ein Wort ein Substantiv, ein Verb oder ein Adjektiv ist. Beziehungen zwischen im Text weit voneinander entfernten Tokens werden erst in tieferen Schichten vorhersagbar (Tenney et al., 2019).

Tiefe Convolutional Neural Networks (CNN) zeigen denselben Verlauf, von Kanten und Texturen in frühen Schichten bis hin zu Objektteilen in tieferen (Zeiler and Fergus, 2014). Vision-Transformer entwickeln ihn ebenfalls: Ihre frühen Schichten teilen die Attention zwischen nahen und fernen Bildausschnitten auf, während die tieferen Schichten das gesamte Bild einbeziehen (Raghu et al., 2021). Sowohl bei Sprach- als auch bei Vision-Transformern entstehen die informativsten Repräsentationen oft in den mittleren Schichten, deren Ausgaben als Merkmale für andere Aufgaben diejenigen der letzten Schicht übertreffen können (Skean et al., 2025).

Was auch immer die Schichten repräsentieren, ihre Berechnung wird von der Token-Mischung dominiert, welche die Matrix jeder Schicht erzeugt, was rechenintensiv ist. Attention vergleicht jedes Paar von Tokens, sodass ihr Aufwand mit der Anzahl n der Tokens wie n^2 wächst (Tay et al., 2022, Abschn. 2.4). Dieses quadratische Wachstum motiviert alternative Techniken zur Token-Mischung. Einige Transformer-Varianten reduzieren die Anzahl der To-

kens in tieferen Schichten (Tay et al., 2022). Jede Transformation, welche die Zeilen von $\mathbf{X}$ kombiniert, kann Attention ersetzen: ein über Tokens hinweg angewandtes MLP (Tolstikhin et al., 2021), eine feste Fourier-Transformation (Lee-Thorp et al., 2022) oder eine Zustandsraum-Schicht, deren Aufwand linear in n ist (Gu and Dao, 2024, Abschn. 3.3).

Siehe auch: Attention, KNN, Schicht, Token, Einbettung, Worteinbettung, RNN, natürliche Sprachverarbeitung (NLP), LLM, MLP, Residualverbindung, Schichtnormalisierung, CNN.

Querverweise

Literatur

1. Brown et al. (2020). Language Models are Few-Shot Learners. In: Adv. Neural Inf. Process. Syst., pp. 1877–1901.
2. Dosovitskiy et al. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: Int. Conf. Learn. Represent. (ICLR).
3. Gu and Dao (2024). Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In: Conf. Language Modeling.
4. Lee-Thorp et al. (2022). FNet: Mixing Tokens with Fourier Transforms. In: Proc. 2022 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technologies, pp. 4296–4313.
5. Grattafiori et al. (2024). The Llama 3 Herd of Models. arXiv preprint arXiv:2407.21783. URL: https://arxiv.org/abs/2407.21783.
6. Raghu et al. (2021). Do Vision Transformers See Like Convolutional Neural Networks?. In: Adv. Neural Inf. Process. Syst., pp. 12116–12128.
7. Skean et al. (2025). Layer by Layer: Uncovering Hidden Representations in Language Models. In: Proc. 42nd Int. Conf. Mach. Learn. (ICML).
8. Tay et al. (2022). Efficient Transformers: A Survey. ACM Computing Surveys, 55(6), pp. Art. no. 109.
9. Tenney et al. (2019). BERT Rediscovers the Classical NLP Pipeline. In: Proc. 57th Ann. Meeting Assoc. Comput. Linguistics, pp. 4593–4601.
10. Tolstikhin et al. (2021). MLP-Mixer: An all-MLP Architecture for Vision. In: Adv. Neural Inf. Process. Syst., pp. 24261–24272.
11. United Nations General Assembly (1948). Universal Declaration of Human Rights (UDHR). United Nations, New York.
12. Zeiler and Fergus (2014). Visualizing and Understanding Convolutional Networks. In: Proc. European Conf. Comput. Vis. (ECCV), pp. 818–833.
13. Vaswani et al. (2017). Attention is all you need. In: Adv. Neural Inf. Process. Syst., pp. 5998–6008.