


Attention Is All You Need

Giảng viên: PhD. Nguyễn Vinh Tiệp


HV: Nguyễn Xuân Trường - 230104027

- Nguyễn Lê Nam Anh - 230101070
- Trần Quốc Huy - 230101048
- Lê Thanh Dũng - 230101074

Tóm tắt nội dung bài báo "Attention Is All You Need"

- Bài báo giới thiệu mô hình Transformer, một kiến trúc mạng mới dựa hoàn toàn vào cơ chế attention, không sử dụng các lớp hồi quy (RNN) hay tích chập (CNN). Transformer đã cho thấy hiệu quả vượt trội trong bài toán dịch máy và các nhiệm vụ xử lý ngôn ngữ tự nhiên khác, với ưu điểm là khả năng song song hóa cao và thời gian huấn luyện ngắn hơn đáng kể so với các mô hình truyền thống.
- 

Tổng quan

- **Mục tiêu:** Đề xuất một kiến trúc mạng mới, Transformer, dựa hoàn toàn vào cơ chế attention, không sử dụng các lớp hồi quy (RNN) hay tích chập (CNN).
 - **Đóng góp:** Chứng minh Transformer vượt trội về chất lượng dịch máy so với các mô hình trước đó, với khả năng song song hóa cao và thời gian huấn luyện ngắn hơn đáng kể.
- 


1. Mô hình dịch chuỗi hiện nay: RNN và CNN

- Recurrent Neural Networks (RNNs) và các biến thể của chúng như Long Short-Term Memory (LSTM) và Gated Recurrent Units (GRU) đã được sử dụng rộng rãi trong các bài toán dịch chuỗi (sequence transduction).
- Convolutional Neural Networks (CNNs) cũng đã được sử dụng để xử lý các chuỗi bằng cách áp dụng các lớp tích chập để nắm bắt các phụ thuộc trong chuỗi.

2. Hạn chế của mô hình hồi quy:

- Khó song song hóa: Do tính tuần tự của RNN, các bước tính toán phải được thực hiện theo thứ tự, làm giảm khả năng song song hóa trong quá trình huấn luyện.
- Giới hạn về độ dài chuỗi: RNN gặp khó khăn khi xử lý các chuỗi dài do vấn đề về gradient (gradient vanishing hoặc exploding), làm giảm khả năng học các phụ thuộc dài hạn trong chuỗi.

3. Giải pháp: Mô hình Transformer dựa hoàn toàn trên attention

- Transformer loại bỏ hoàn toàn các lớp hồi quy và tích chập, thay vào đó sử dụng cơ chế attention để xử lý các chuỗi.
 - Ưu điểm của Transformer:
 - Khả năng song song hóa tốt hơn, giúp giảm thời gian huấn luyện.
 - Khả năng học các phụ thuộc dài hạn tốt hơn nhờ cơ chế attention.
- 

Attention là gì

- **Định nghĩa Attention:**
 - Attention là một cơ chế trong các mô hình học sâu, đặc biệt là trong các bài toán xử lý ngôn ngữ tự nhiên và dịch máy.
 - Nó cho phép mô hình tập trung vào các phần quan trọng của chuỗi đầu vào khi tạo ra chuỗi đầu ra, thay vì xử lý toàn bộ chuỗi một cách đồng đều.
- **Cách hoạt động của Attention:**
 - Truy vấn (Query), Khóa (Key), và Giá trị (Value):
 - ✓ Attention nhận vào một tập các truy vấn, khóa và giá trị.
 - ✓ Truy vấn đại diện cho từ hiện tại đang được xử lý.
 - ✓ Khóa và giá trị đại diện cho tất cả các từ trong chuỗi đầu vào.
 - Tính toán trọng số:
 - ✓ Tích vô hướng (dot product) giữa truy vấn và mỗi khóa để tính độ tương quan.
 - ✓ Sử dụng hàm softmax để chuyển đổi các giá trị tương quan này thành các trọng số xác suất.
 - Tính toán đầu ra:
 - ✓ Trọng số này được nhân với các giá trị tương ứng và cộng lại để tạo ra đầu ra cho từ hiện tại.
- **Ưu điểm của Attention:**
 - Phụ thuộc dài hạn:
 - ✓ Attention giúp mô hình học các phụ thuộc giữa các từ ở khoảng cách xa trong chuỗi, khắc phục hạn chế của các mô hình hồi quy truyền thống.
 - Xử lý chuỗi biến đổi:
 - ✓ Attention không bị giới hạn bởi độ dài chuỗi, cho phép xử lý các chuỗi có độ dài biến đổi một cách hiệu quả.
 - Hiệu quả tính toán:
 - ✓ Attention cho phép song song hóa trong quá trình huấn luyện, tăng tốc độ tính toán và giảm thời gian huấn luyện.

Self-Attention

➤ Định nghĩa Self-Attention:

- Self-Attention, hay còn gọi là Intra-Attention, là một dạng attention mà trong đó các từ trong chuỗi đầu vào tự liên hệ với nhau để tính toán biểu diễn của chuỗi.

➤ Cách hoạt động của Self-Attention:

- Tính toán trọng số:
 - ✓ Mỗi từ trong chuỗi đầu vào được sử dụng làm truy vấn (query) và tính toán độ tương quan với tất cả các từ khác trong chuỗi.
 - ✓ Các trọng số tương quan này sau đó được sử dụng để tính toán một biểu diễn mới cho từ truy vấn dựa trên các từ còn lại trong chuỗi.

➤ Tổng hợp thông tin:

- Biểu diễn mới của từ truy vấn được tổng hợp từ biểu diễn của tất cả các từ khác trong chuỗi, dựa trên các trọng số đã tính toán.

➤ Lợi ích của Self-Attention:

- Phụ thuộc dài hạn:
 - ✓ Self-Attention giúp mô hình dễ dàng học các phụ thuộc giữa các từ ở khoảng cách xa, cải thiện khả năng hiểu ngữ cảnh toàn cục của chuỗi.
- Khả năng song song hóa:
 - ✓ Self-Attention cho phép xử lý toàn bộ chuỗi đầu vào đồng thời, giúp tăng tốc độ tính toán và giảm thời gian huấn luyện.

Multi-Head Attention

➤ Định nghĩa Multi-Head Attention:

- Multi-Head Attention là kỹ thuật mở rộng attention bằng cách sử dụng nhiều "đầu" attention hoạt động song song, giúp mô hình có thể học được nhiều khía cạnh khác nhau của dữ liệu đầu vào.

➤ Cách hoạt động của Multi-Head Attention:

- Phân chia không gian:
 - ✓ Đầu tiên, các vector đầu vào được chia thành nhiều không gian con khác nhau.
- Attention đa đầu:
 - ✓ Trên mỗi không gian con, một đầu attention riêng biệt được áp dụng để tính toán trọng số và tổng hợp thông tin.
- Kết hợp kết quả:
 - ✓ Kết quả từ tất cả các đầu attention được kết hợp lại để tạo thành một biểu diễn cuối cùng, giúp nắm bắt thông tin đa dạng và phong phú hơn.

➤ Lợi ích của Multi-Head Attention:

- Nắm bắt thông tin đa chiều:
 - ✓ Multi-Head Attention cho phép mô hình nắm bắt thông tin từ nhiều khía cạnh khác nhau, cải thiện khả năng hiểu ngữ cảnh và mối quan hệ giữa các từ trong chuỗi.

➤ Hiệu suất và biểu diễn:

- Việc sử dụng nhiều đầu attention giúp cải thiện hiệu suất và khả năng biểu diễn của mô hình, đồng thời giúp mô hình học được các đặc trưng phức tạp hơn của dữ liệu.

Kiến trúc mô hình

I. Encoder:

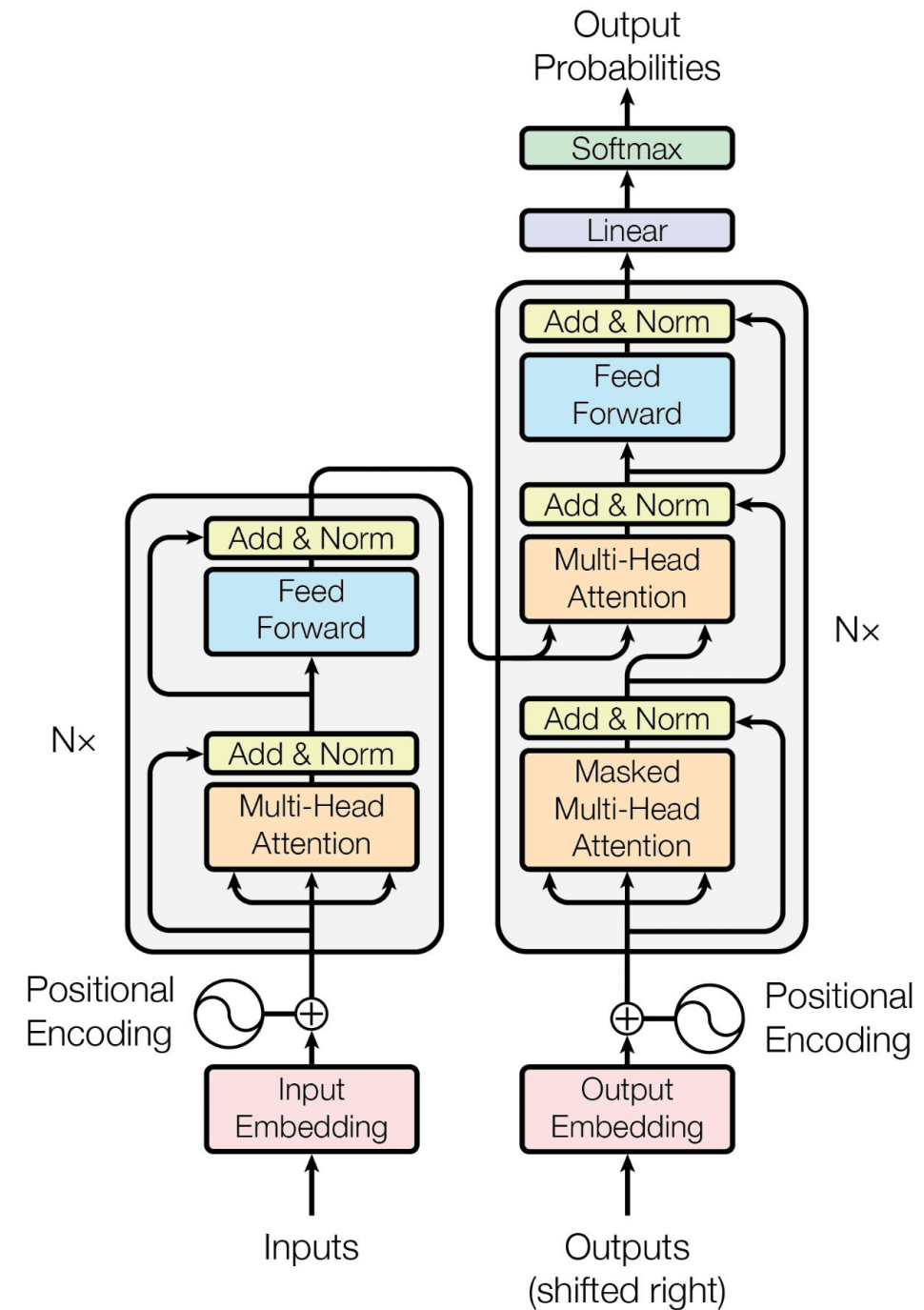
1. Gồm 6 lớp tương tự nhau ($N=6$).
2. Mỗi lớp bao gồm:
 - Multi-head self-attention: Học các mối quan hệ giữa các từ trong câu.
 - Feed-forward network: Áp dụng trên từng vị trí riêng lẻ.
 - Residual connection và layer normalization: Kết nối dư và chuẩn hóa lớp để duy trì tính ổn định.

II. Decoder:

1. Gồm 6 lớp tương tự nhau ($N=6$).
2. Mỗi lớp bao gồm:
 - Self-attention: Tự chú ý vào các từ trong chuỗi đầu ra, nhưng ngăn cản việc chú ý đến các vị trí phía sau.
 - Encoder-decoder attention: Chú ý vào các từ trong chuỗi đầu vào của encoder.
 - Feed-forward network: Áp dụng trên từng vị trí riêng lẻ.
 - Residual connection và layer normalization: Kết nối dư và chuẩn hóa lớp để duy trì tính ổn định.

III. Các thành phần khác:

1. Positional Encoding: Sử dụng hàm sin và cosin để mã hóa vị trí của từ trong chuỗi.
2. Output Linear và Softmax: Chuyển đổi đầu ra của decoder thành xác suất của các từ tiếp theo.



Lợi ích của Self-Attention

1. Khả năng học các phụ thuộc dài hạn:

- Self-Attention cho phép mỗi từ trong chuỗi đầu vào xem xét toàn bộ các từ khác trong chuỗi, giúp mô hình học được các phụ thuộc dài hạn và ngữ cảnh toàn diện hơn.
- Điều này đặc biệt hữu ích trong các bài toán xử lý ngôn ngữ tự nhiên, nơi mà các từ ở xa nhau trong câu có thể có mối quan hệ chặt chẽ.

2. Tăng cường khả năng song song hóa:

- Khác với các mô hình hồi quy, Self-Attention cho phép tính toán song song, xử lý toàn bộ chuỗi đầu vào cùng một lúc.
- Điều này giúp tăng tốc độ huấn luyện và tận dụng tối đa tài nguyên phần cứng hiện đại.


3. Tính toán hiệu quả:

- Độ phức tạp tính toán của Self-Attention là $O(n^2)$, thấp hơn so với các mô hình hồi quy có độ phức tạp là $O(n)$.
- Self-Attention giúp tiết kiệm tài nguyên và cải thiện hiệu suất tính toán.


4. Mô hình linh hoạt:

- Self-Attention không bị giới hạn bởi độ dài chuỗi, cho phép xử lý các chuỗi có độ dài biến đổi một cách hiệu quả.
- Điều này làm cho mô hình trở nên linh hoạt hơn trong việc áp dụng vào các bài toán thực tế với dữ liệu đầu vào đa dạng.

Phương pháp thử nghiệm

- **Dữ liệu:** Sử dụng bộ dữ liệu WMT 2014 cho dịch Anh-Đức và Anh-Pháp.
 - **Phần cứng:** Huấn luyện trên máy có 8 GPU NVIDIA P100.
 - **Tối ưu hóa:** Sử dụng Adam optimizer với các tham số $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1e-9$.
 - **Regularization:** Áp dụng dropout và label smoothing.
- 

Kết quả

- Trên tác vụ dịch máy từ Anh sang Đức và từ Anh sang Pháp, mô hình Transformer đạt được điểm BLEU cao hơn so với các mô hình state-of-the-art trước đó.
 - Cụ thể, trên tác vụ dịch Anh-Đức, Transformer đạt 28.4 BLEU, cao hơn 2.0 BLEU so với mô hình tốt nhất trước đó.
 - Trên tác vụ dịch Anh-Pháp, Transformer đạt 41.8 BLEU, cũng cao hơn đáng kể so với các mô hình khác.
- 

Phân tích

Mô hình	BLEU (EN-DE)	BLEU (EN-FR)	Chi phí huấn luyện (FLOPs)
ByteNet	23.75	-	-
Deep-Att + PosUnk	-	39.2	1.0×10^{20}
GNMT + RL	24.6	39.92	2.3×10^{19} (EN-DE), 1.4×10^{20} (EN-FR)
ConvS2S	25.16	40.46	9.6×10^{18} (EN-DE), 1.5×10^{20} (EN-FR)
MoE	26.03	40.56	2.0×10^{19} (EN-DE), 1.2×10^{20} (EN-FR)
Deep-Att + PosUnk Ensemble	-	40.4	8.0×10^{20}
GNMT + RL Ensemble	26.30	41.16	1.8×10^{20} (EN-DE), 1.1×10^{21} (EN-FR)
ConvS2S Ensemble	26.36	41.29	7.7×10^{19} (EN-DE), 1.2×10^{21} (EN-FR)
Transformer (base model)	27.3	38.1	3.3×10^{18}
Transformer (big model)	28.4	41.8	2.3×10^{19}

Lợi ích và hạn chế của mô hình Transformer

- Lợi ích:

- Khả năng học tương quan dài hạn
- Hiệu suất cao trong xử lý ngôn ngữ tự nhiên
- Có thể áp dụng cho nhiều tác vụ khác nhau
- Tính Parallelizable cao

- Hạn chế:

- Yêu cầu tài nguyên tính toán lớn
- Độ phức tạp cao
- Dễ bị overfitting
- Khó khăn trong việc giải thích

Tài liệu tham khảo

