Machine Translated by Google

Với điều kiện ghi rõ nguồn, Google sẽ cấp phép sao chép các bảng và hình ảnh trong bài viết này chỉ để sử dụng trong các tác phẩm báo chí hoặc học thuật.

Sự chú ý là tất cả những gì bạn cần

Ashish Vaswani
Google Brain
avaswani@google.com

Noam Shazeer
Google Brain
noam@google.com

Niki Parmar
Nghiên cứu của Google
nikip@google.com

Jakob Uszkoreit
Nghiên cứu của Google
usz@google.com

Llion Jones
Google Nghiên cứu
llion@google.com

Aidan N. Gomez †
Đại học Toronto
aidan@cs.toronto.edu

Łukasz Kaiser
Google Brain
lukaszkaiser@google.com

Illia Polosukhin
‡ illia.polosukhin@gmail.com

Tóm tắt

Các mô hình chuyển đổi chuỗi thống trị dựa trên các mạng nơ-ron hồi quy hoặc tích chập phức tạp bao gồm một bộ mã hóa và một bộ giải mã. Các mô hình có hiệu suất tốt nhất cũng kết nối bộ mã hóa và giải mã thông qua một cơ chế chú ý. Chúng tôi đề xuất một kiến trúc mạng đơn giản mới, Transformer, chỉ dựa trên các cơ chế chú ý, loại bỏ hoàn toàn sự hồi quy và tích chập. Các thí nghiệm trên hai tác vụ dịch máy cho thấy các mô hình này có chất lượng vượt trội trong khi có khả năng song song hóa cao hơn và cần ít thời gian đào tạo hơn đáng kể. Mô hình của chúng tôi đạt 28,4 BLEU trên tác vụ dịch từ tiếng Anh sang tiếng Đức WMT 2014, cải thiện hơn 2 BLEU so với các kết quả tốt nhất hiện có, bao gồm cả các tập hợp. Trên tác vụ dịch từ tiếng Anh sang tiếng Pháp WMT 2014, mô hình của chúng tôi thiết lập một điểm BLEU hiện đại mới của mô hình đơn là 41,8 sau khi đào tạo trong 3,5 ngày trên tám GPU, một phần nhỏ chi phí đào tạo của các mô hình tốt nhất trong tài liệu. Chúng tôi chứng minh rằng Transformer có thể khái quát hóa tốt cho các tác vụ khác bằng cách áp dụng thành công vào việc phân tích cú pháp thành phần tiếng Anh với cả dữ liệu đào tạo lớn và hạn chế.

[cs.CL] 1706.03762v7

Đóng góp ngang nhau. Thứ tự liệt kê là ngẫu nhiên. Jakob đề xuất thay thế RNN bằng self-attention và bắt đầu nỗ lực đánh giá ý tưởng này. Ashish, cùng với Illia, đã thiết kế và triển khai các mô hình Transformer đầu tiên và đã tham gia rất quan trọng vào mọi khía cạnh của công trình này. Noam đề xuất sự chú ý tích vô hướng tỷ lệ, sự chú ý đa đầu và biểu diễn vị trí không tham số và trở thành người thứ hai tham gia vào hầu hết mọi chi tiết. Niki đã thiết kế, triển khai, điều chỉnh và đánh giá vô số biến thể mô hình trong cơ sở mã gốc và tensor2tensor của chúng tôi. Llion cũng đã thử nghiệm với các biến thể mô hình mới, chịu trách nhiệm cho cơ sở mã ban đầu của chúng tôi và suy luận và trực quan hóa hiệu quả. Lukasz và Aidan đã dành vô số ngày dài để thiết kế các phần khác nhau của tensor2tensor và triển khai, thay thế cơ sở mã trước đó của chúng tôi, cải thiện đáng kể kết quả và đẩy nhanh đáng kể quá trình nghiên cứu của chúng tôi.

†Công việc được thực hiện khi làm việc tại Google Brain. ‡Công việc được thực hiện khi làm việc tại Google Research.

Hội nghị lần thứ 31 về Hệ thống xử lý thông tin thần kinh (NIPS 2017), Long Beach, CA, Hoa Kỳ.

Machine Translated by Google

1 Giới thiệu

Mạng nơ-ron hồi quy, đặc biệt là mạng nơ-ron bộ nhớ dài hạn ngắn [13] và mạng nơ-ron hồi quy có cổng [7] , đã được khẳng định là phương pháp tiếp cận tiên tiến trong các vấn đề mô hình hóa trình tự và chuyển đổi như mô hình hóa ngôn ngữ và dịch máy [35, 2, 5]. Nhiều nỗ lực kể từ đó đã tiếp tục thúc đẩy ranh giới của các mô hình ngôn ngữ hồi quy và kiến trúc mã hóa-giải mã [38, 24, 15].

Các mô hình hồi quy thường phân tích tính toán theo vị trí ký hiệu của chuỗi đầu vào và đầu ra. Bằng cách căn chỉnh các vị trí theo các bước trong thời gian tính toán, chúng tạo ra một chuỗi các trạng thái ẩn ht, theo hàm của trạng thái ẩn ht 1 trước đó và đầu vào cho vị trí t. Bản chất tuần tự vốn có này ngăn cản việc song song hóa trong các ví dụ huấn luyện, điều này trở nên quan trọng ở các chuỗi dài hơn, vì các hạn chế về bộ nhớ giới hạn việc xử lý hàng loạt trên các ví dụ. Các nghiên cứu gần đây đã đạt được những cải tiến đáng kể về hiệu quả tính toán thông qua các thủ thuật phân tích [21] và tính toán có điều kiện [32], đồng thời cũng cải thiện hiệu suất mô hình trong trường hợp sau. Tuy nhiên, hạn chế cơ bản của tính toán tuần tự vẫn còn.

Cơ chế chú ý đã trở thành một phần không thể thiếu của mô hình hóa trình tự hấp dẫn và các mô hình chuyển đổi trong nhiều nhiệm vụ khác nhau, cho phép mô hình hóa các mối phụ thuộc mà không cần quan tâm đến khoảng cách của chúng trong trình tự đầu vào hoặc đầu ra [2, 19]. Tuy nhiên, trong hầu hết các trường hợp [27], các cơ chế chú ý như vậy được sử dụng kết hợp với mạng lưới tuần hoàn.

Trong công trình này, chúng tôi đề xuất Transformer, một kiến trúc mô hình tránh sự lặp lại và thay vào đó hoàn toàn dựa vào cơ chế chú ý để rút ra các mối phụ thuộc toàn cục giữa đầu vào và đầu ra. Transformer cho phép song song hóa nhiều hơn đáng kể và có thể đạt đến trạng thái nghệ thuật mới về chất lượng dịch thuật sau khi được đào tạo chỉ trong mười hai giờ trên tám GPU P100.

2 Bối cảnh

Mục tiêu giảm tính toán tuần tự cũng hình thành nền tảng của GPU thần kinh mở rộng [16], ByteNet [18] và ConvS2S [9], tất cả đều sử dụng mạng nơ-ron tích chập làm khối xây dựng cơ bản, tính toán các biểu diễn ẩn song song cho tất cả các vị trí đầu vào và đầu ra. Trong các mô hình này, số lượng hoạt động cần thiết để liên hệ các tín hiệu từ hai vị trí đầu vào hoặc đầu ra tùy ý tăng theo khoảng cách giữa các vị trí, tuyến tính đối với ConvS2S và logarit đối với ByteNet. Điều này làm cho việc học các mối quan hệ phụ thuộc giữa các vị trí xa nhau trở nên khó khăn hơn [12]. Trong Transformer, điều này được giảm xuống thành một số lượng hoạt động không đổi, mặc dù phải trả giá bằng độ phân giải hiệu quả giảm do tính trung bình các vị trí có trọng số chú ý, một hiệu ứng mà chúng tôi chống lại bằng Chú ý nhiều đầu như được mô tả trong phần 3.2.

Tự chú ý, đôi khi được gọi là nội chú ý, là một cơ chế chú ý liên hệ các vị trí khác nhau của một chuỗi duy nhất để tính toán biểu diễn của chuỗi đó. Tự chú ý đã được sử dụng thành công trong nhiều nhiệm vụ khác nhau, bao gồm hiểu đọc, tóm tắt trừu tượng, suy diễn văn bản và biểu diễn câu độc lập với nhiệm vụ học tập [4, 27, 28, 22].

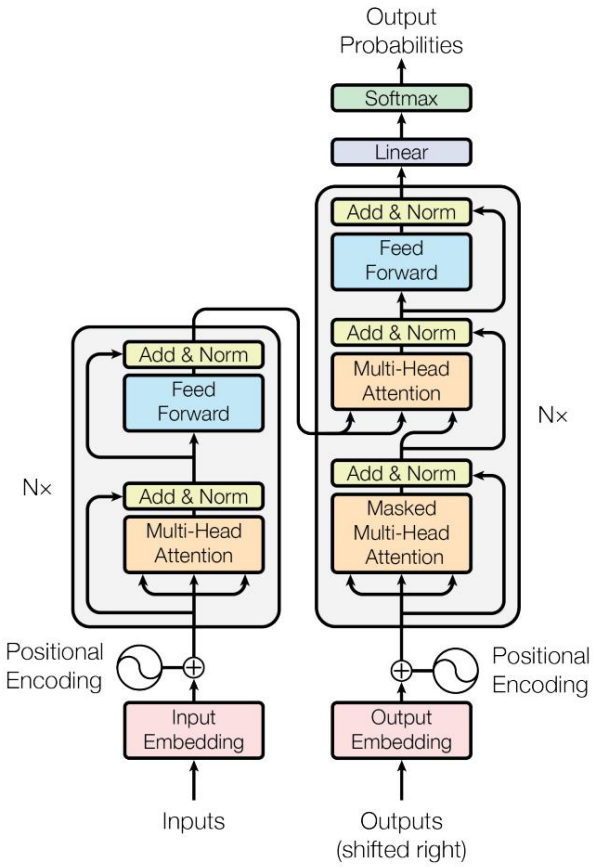
Mạng bộ nhớ đầu cuối dựa trên cơ chế chú ý tuần hoàn thay vì tuần hoàn theo trình tự và đã được chứng minh là hoạt động tốt trong các nhiệm vụ trả lời câu hỏi ngôn ngữ đơn giản và mô hình hóa ngôn ngữ [34].

Tuy nhiên, theo hiểu biết của chúng tôi, Transformer là mô hình chuyển đổi đầu tiên hoàn toàn dựa vào sự tự chú ý để tính toán biểu diễn đầu vào và đầu ra mà không sử dụng RNN được căn chỉnh theo trình tự hoặc tích chập. Trong các phần sau, chúng tôi sẽ mô tả Transformer, thúc đẩy sự tự chú ý và thảo luận về những ưu điểm của nó so với các mô hình như [17, 18] và [9].

3 Kiến trúc mô hình

Hầu hết các mô hình chuyển đổi chuỗi nơ-ron cạnh tranh đều có cấu trúc mã hóa-giải mã [5, 2, 35]. Ở đây, bộ mã hóa ánh xạ một chuỗi đầu vào của các biểu diễn ký hiệu (x1, ..., xn) thành một chuỗi các biểu diễn liên tục z = (z1, ..., zn). Với z, bộ giải mã sau đó tạo ra một chuỗi đầu ra (y1, ..., ym) của các ký hiệu, mỗi lần một phần tử. Ở mỗi bước, mô hình tự hồi quy [10], sử dụng các ký hiệu đã tạo trước đó làm đầu vào bổ sung khi tạo ra ký hiệu tiếp theo.

Machine Translated by Google



Hình 1: Kiến trúc mô hình của Transformer.

Transformer tuân theo kiến trúc tổng thể này bằng cách sử dụng các lớp tự chú ý xếp chồng và từng điểm, được kết nối đầy đủ cho cả bộ mã hóa và bộ giải mã, được hiển thị ở nửa bên trái và bên phải của Hình 1 .

3.1 Bộ mã hóa và bộ giải mã

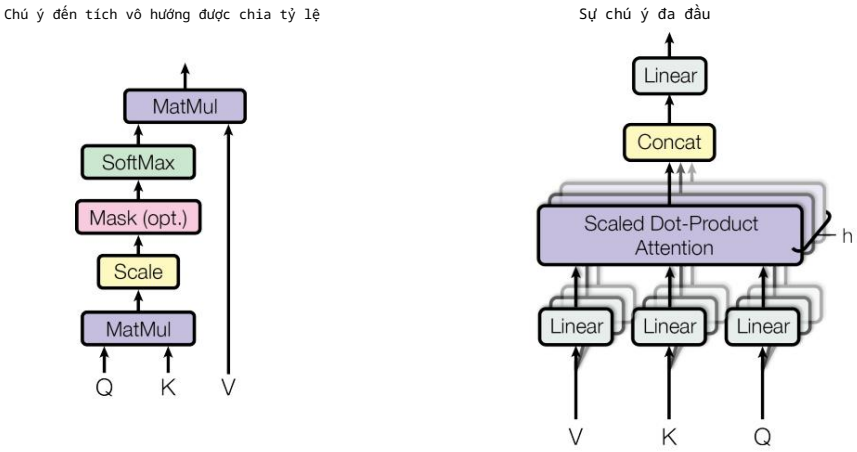
Bộ mã hóa: Bộ mã hóa được tạo thành từ một chồng N = 6 lớp giống hệt nhau. Mỗi lớp có hai lớp con. Lớp đầu tiên là cơ chế tự chú ý đa đầu, và lớp thứ hai là một mạng truyền thẳng hoàn toàn, kết nối đầy đủ theo vị trí. Chúng tôi sử dụng một kết nối dư [11] xung quanh mỗi lớp con trong hai lớp con, tiếp theo là chuẩn hóa lớp [1]. Nghĩa là, đầu ra của mỗi lớp con là LayerNorm(x + Sublayer(x)), trong đó Sublayer(x) là hàm được triển khai bởi chính lớp con đó. Để tạo điều kiện cho các kết nối dư này, tất cả các lớp con trong mô hình, cũng như các lớp nhúng, tạo ra đầu ra có chiều dmodel = 512.

Bộ giải mã: Bộ giải mã cũng bao gồm một chồng N = 6 lớp giống hệt nhau. Ngoài hai lớp con trong mỗi lớp mã hóa, bộ giải mã còn chèn thêm một lớp con thứ ba, thực hiện chức năng chú ý đa đầu (multi-head attention) trên đầu ra của chồng mã hóa. Tương tự như bộ mã hóa, chúng tôi sử dụng các kết nối dư (residual connections) xung quanh mỗi lớp con, sau đó là chuẩn hóa lớp. Chúng tôi cũng điều chỉnh lớp con tự chú ý trong chồng giải mã để ngăn các vị trí không chú ý đến các vị trí tiếp theo. Việc che dấu này, kết hợp với việc các nhúng đầu ra được bù trừ một vị trí, đảm bảo rằng các dự đoán cho vị trí i chỉ có thể phụ thuộc vào các đầu ra đã biết ở các vị trí nhỏ hơn i.

3.2 Chú ý

Hàm chú ý có thể được mô tả như một phép ánh xạ một truy vấn và một tập hợp các cặp khóa-giá trị đến một đầu ra, trong đó truy vấn, khóa, giá trị và đầu ra đều là các vectơ. Đầu ra được tính dưới dạng tổng có trọng số.

Machine Translated by Google



Hình 2: (trái) Chú ý tích vô hướng được chia tỷ lệ. (phải) Chú ý nhiều đầu bao gồm một số lớp chú ý chạy song song.

của các giá trị, trong đó trọng số được gán cho mỗi giá trị được tính bằng hàm tương thích của truy vấn với khóa tương ứng.

3.2.1 Tích vô hướng tỷ lệ Chú ý

Chúng tôi gọi sự chú ý đặc biệt của mình là "Sự chú ý tích vô hướng tỉ lệ" (Hình 2). Đầu vào bao gồm các truy vấn và khóa có chiều dk, và các giá trị có chiều dv. Chúng tôi tính tích vô hướng của truy vấn với tất cả các khóa, chia mỗi khóa cho $\sqrt{dk}$, và áp dụng hàm softmax để thu được trọng số trên các giá trị.

Trong thực tế, chúng tôi tính toán hàm chú ý trên một tập hợp các truy vấn đồng thời, được đóng gói lại thành ma trận Q. Các khóa và giá trị cũng được đóng gói lại thành ma trận K và V. Chúng tôi tính toán ma trận đầu ra như sau:

$$\text{Chú ý}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{dk}})V \quad (1)$$

Hai hàm chú ý được sử dụng phổ biến nhất là chú ý cộng tính [2] và chú ý tích vô hướng (nhân). Chú ý tích vô hướng giống hệt thuật toán của chúng tôi, ngoại trừ hệ số tỷ lệ là $\frac{1}{\sqrt{dk}}$. Chú ý cộng tính toán hàm tương thích bằng cách sử dụng mạng truyền thẳng với một lớp ẩn duy nhất. Mặc dù cả hai có độ phức tạp lý thuyết tương đương nhau, nhưng chú ý tích vô hướng nhanh hơn nhiều và tiết kiệm không gian hơn trong thực tế, vì nó có thể được triển khai bằng mã nhân ma trận được tối ưu hóa cao.

Trong khi đối với các giá trị nhỏ của dk, hai cơ chế hoạt động tương tự nhau, sự chú ý cộng tính vượt trội hơn sự chú ý tích vô hướng mà không cần mở rộng đối với các giá trị lớn hơn của dk [3]. Chúng tôi nghi ngờ rằng đối với các giá trị lớn của dk, các tích vô hướng tăng trưởng lớn về độ lớn, đẩy hàm softmax vào các vùng có độ dốc cực nhỏ. [4] Để chống lại hiệu ứng này, chúng ta chia tỷ lệ các tích vô hướng theo $\frac{1}{\sqrt{dk}}$.

3.2.2 Sự chú ý đa đầu

Thay vì thực hiện một hàm chú ý duy nhất với các khóa, giá trị và truy vấn theo chiều dmodel, chúng tôi thấy việc chiếu tuyến tính các truy vấn, khóa và giá trị h lần với các phép chiếu tuyến tính đã học khác nhau lên các chiều dk, dk và dv tương ứng sẽ có lợi hơn. Trên mỗi phiên bản được chiếu của các truy vấn, khóa và giá trị này, chúng tôi thực hiện hàm chú ý song song, tạo ra các chiều dv.

[4] Để minh họa lý do tại sao các tích vô hướng trở nên lớn, hãy giả sử rằng các thành phần của q và k là qiki ngẫu nhiên độc lập, có giá trị trung bình là 0 và phương sai là 1. Khi đó tích vô hướng của chúng, $q \cdot k = \sum_{i=1}^{dk} qiki$, có giá trị trung bình là 0 và phương sai là dk.

Machine Translated by Google

giá trị đầu ra. Chúng được nối lại và một lần nữa được chiếu, tạo ra các giá trị cuối cùng, như được mô tả trong Hình 2.

Sự chú ý đa đầu cho phép mô hình cùng lúc xử lý thông tin từ các không gian biểu diễn khác nhau ở các vị trí khác nhau. Với một đầu chú ý duy nhất, việc tính trung bình sẽ hạn chế điều này.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \text{ trong đó head}_i$$

$$= \text{Attention}(QW^Q_i, KW^K_i, VW^V_i)$$

Trong đó các phép chiếu là ma trận tham số $W^Q_i \in \mathbb{R}^{d_{model} \times d_k}$, $W^K_i \in \mathbb{R}^{d_{model} \times d_k}$, $W^V_i \in \mathbb{R}^{d_{model} \times d_v}$ và $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

Trong nghiên cứu này, chúng tôi sử dụng h = 8 lớp chú ý song song, hay còn gọi là đầu. Với mỗi lớp, chúng tôi sử dụng dk = dv = dmodel/h = 64. Do kích thước của mỗi đầu được thu nhỏ, tổng chi phí tính toán tương tự như khi sử dụng chú ý một đầu với đầy đủ chiều.

3.2.3 Ứng dụng của sự chú ý trong mô hình của chúng tôi

Transformer sử dụng sự chú ý của nhiều đầu theo ba cách khác nhau:

- Trong các lớp "chú ý mã hóa-giải mã", các truy vấn đến từ lớp giải mã trước đó, còn các khóa và giá trị bộ nhớ đến từ đầu ra của bộ mã hóa. Điều này cho phép mọi vị trí trong bộ giải mã đều có thể xử lý tất cả các vị trí trong chuỗi đầu vào. Điều này mô phỏng các cơ chế chú ý mã hóa-giải mã điển hình trong các mô hình chuỗi-đến-chuỗi như [38, 2, 9].

- Bộ mã hóa chứa các lớp tự chú ý. Trong một lớp tự chú ý, tất cả các khóa, giá trị và truy vấn đều đến từ cùng một nơi, trong trường hợp này là đầu ra của lớp trước đó trong bộ mã hóa. Mỗi vị trí trong bộ mã hóa có thể tương ứng với tất cả các vị trí trong lớp trước đó của bộ mã hóa.

- Tương tự, các lớp tự chú ý trong bộ giải mã cho phép mỗi vị trí trong bộ giải mã chú ý đến tất cả các vị trí trong bộ giải mã, bao gồm cả vị trí đó. Chúng ta cần ngăn chặn luồng thông tin sang trái trong bộ giải mã để duy trì tính chất tự hồi quy. Chúng tôi triển khai điều này bên trong sự chú ý tích vô hướng tỷ lệ bằng cách che giấu (đặt thành -∞) tất cả các giá trị trong đầu vào của softmax tương ứng với các kết nối bất hợp pháp. Xem Hình 2.

3.3 Mạng truyền thẳng theo vị trí

Ngoài các lớp phụ chú ý, mỗi lớp trong bộ mã hóa và giải mã của chúng tôi đều chứa một mạng truyền thẳng hoàn toàn được kết nối, được áp dụng cho từng vị trí riêng biệt và giống hệt nhau. Mạng này bao gồm hai phép biến đổi tuyến tính với một phép kích hoạt ReLU ở giữa.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2)$$

Mặc dù các phép biến đổi tuyến tính giống nhau ở các vị trí khác nhau, chúng sử dụng các tham số khác nhau từ lớp này sang lớp khác. Một cách khác để mô tả điều này là hai phép tích chập với kích thước hạt nhân là 1.
Số chiều của đầu vào và đầu ra là dmodel = 512 và lớp bên trong có số chiều df f = 2048.

3.4 Nhúng và Softmax

Tương tự như các mô hình chuyển đổi chuỗi khác, chúng tôi sử dụng các phép nhúng đã học để chuyển đổi các token đầu vào và token đầu ra thành các vectơ có chiều dmodel. Chúng tôi cũng sử dụng phép biến đổi tuyến tính đã học thông thường và hàm softmax để chuyển đổi đầu ra của bộ giải mã thành các xác suất token tiếp theo được dự đoán. Trong mô hình của chúng tôi, chúng tôi chia sẻ cùng một ma trận trọng số giữa hai lớp nhúng và phép biến đổi tuyến tính tiền softmax, tương tự như [30]. Trong các lớp nhúng, chúng tôi nhân các trọng số đó với $\sqrt{d_{model}}$.

Machine Translated by Google

Bảng 1: Độ dài đường dẫn tối đa, độ phức tạp của mỗi lớp và số lượng thao tác tuần tự tối thiểu cho các loại lớp khác nhau. n là độ dài chuỗi, d là chiều biểu diễn, k là hạt nhân kích thước của các phép tích chập và r là kích thước của vùng lân cận trong sự chú ý hạn chế.

Loại lớp	Độ phức tạp của mỗi lớp	Chiều dài đường dẫn tối đa tuần tự Hoạt động	
Tự chú ý	$O(n^2 \cdot d)$	O(1)	O(1)
Định kỳ	$O(n \cdot d^2)$	O(n)	O(n)
Tích chập	$O(k \cdot n \cdot d^2)$	O(1)	O(logk(n))
Tự chú ý (có giới hạn)	O(r · n · d)	O(1)	O(n/r)

3.5 Mã hóa vị trí

Vì mô hình của chúng tôi không chứa sự lặp lại và không có tích chập, để mô hình có thể sử dụng thứ tự của chuỗi, chúng ta phải đưa vào một số thông tin về vị trí tương đối hoặc tuyệt đối của mã thông báo trong chuỗi. Để đạt được mục đích này, chúng tôi thêm "mã hóa vị trí" vào nhúng đầu vào tại đáy của ngăn xếp bộ mã hóa và bộ giải mã. Các mã hóa vị trí có cùng kích thước dmodel như các nhúng, để hai phần này có thể được cộng lại. Có nhiều lựa chọn mã hóa vị trí, đã học và cố định [9].

Trong công trình này, chúng tôi sử dụng các hàm sin và cos có tần số khác nhau:

$$P E(pos,2i) = \sin(pos/100002i/dmodel)$$

$$P E(pos,2i+1) = \cos(pos/100002i/dmodel)$$

trong đó pos là vị trí và i là chiều. Nghĩa là, mỗi chiều của mã hóa vị trí tương ứng với một hình sin. Các bước sóng tạo thành một cấp số nhân từ 2π đến 10000 · 2π. Chúng ta đã chọn chức năng này vì chúng tôi đưa ra giả thuyết rằng nó sẽ cho phép mô hình dễ dàng học cách tham dự bằng vị trí tương đối, vì đối với bất kỳ độ lệch cố định k nào, P Epos+k có thể được biểu diễn dưới dạng hàm tuyến tính của P Sử thi

Chúng tôi cũng đã thử nghiệm bằng cách sử dụng các nhúng vị trí đã học [9] thay thế và thấy rằng hai các phiên bản tạo ra kết quả gần như giống hệt nhau (xem Bảng 3 hàng (E)). Chúng tôi đã chọn phiên bản hình sin bởi vì nó có thể cho phép mô hình ngoại suy đến độ dài chuỗi dài hơn những chuỗi gặp phải trong quá trình đào tạo.

4 Tại sao phải tự chú ý

Trong phần này, chúng tôi so sánh các khía cạnh khác nhau của các lớp tự chú ý với các lớp tuần hoàn và lớp tích chập thường được sử dụng để ánh xạ một chuỗi biểu diễn ký hiệu có độ dài thay đổi (x1, ..., xn) thành một chuỗi khác có độ dài bằng nhau (z1, ..., zn), với xi ,zi R ℵ, chẳng hạn như một ẩn lớp trong một bộ mã hóa hoặc giải mã chuyển đổi trình tự điển hình. Thúc đẩy việc sử dụng sự tự chú ý của chúng ta xem xét ba điều mong muốn.

Một là tổng độ phức tạp tính toán trên mỗi lớp. Một là lượng tính toán có thể được song song hóa, được đo bằng số lượng tối thiểu các phép toán tuần tự cần thiết.

Thứ ba là độ dài đường dẫn giữa các phụ thuộc tầm xa trong mạng. Học tầm xa sự phụ thuộc là một thách thức quan trọng trong nhiều nhiệm vụ chuyển đổi trình tự. Một yếu tố quan trọng ảnh hưởng đến khả năng học các phụ thuộc như vậy là độ dài của các đường dẫn tới và lui mà tín hiệu phải có đi qua trong mạng. Các đường dẫn này giữa bất kỳ tổ hợp vị trí nào trong đầu vào càng ngắn và các chuỗi đầu ra, thì việc học các phụ thuộc tầm xa càng dễ dàng hơn [12]. Do đó, chúng tôi cũng so sánh độ dài đường dẫn tối đa giữa bất kỳ hai vị trí đầu vào và đầu ra nào trong các mạng được tạo thành từ các loại lớp khác nhau.

Như đã lưu ý trong Bảng 1, một lớp tự chú ý kết nối tất cả các vị trí với một số lượng không đổi tuần tự các hoạt động được thực hiện, trong khi một lớp tuần hoàn yêu cầu O(n) hoạt động tuần tự. Về mặt độ phức tạp tính toán, các lớp tự chú ý nhanh hơn các lớp tuần hoàn khi trình tự

Machine Translated by Google

Độ dài n nhỏ hơn chiều biểu diễn d, điều này thường xảy ra với các biểu diễn câu được sử dụng bởi các mô hình tiên tiến trong dịch máy, chẳng hạn như biểu diễn từng từ [38] và từng cặp byte [31] . Để cải thiện hiệu suất tính toán cho các tác vụ liên quan đến chuỗi rất dài, sự tự chú ý có thể được giới hạn trong việc chỉ xem xét một vùng lân cận có kích thước r trong chuỗi đầu vào tập trung xung quanh vị trí đầu ra tương ứng. Điều này sẽ tăng độ dài đường dẫn tối đa lên O(n/r). Chúng tôi dự định sẽ nghiên cứu sâu hơn về phương pháp này trong các công trình tương lai.

Một lớp tích chập đơn với chiều rộng hạt nhân k < n không kết nối tất cả các cặp vị trí đầu vào và đầu ra. Làm như vậy đòi hỏi một ngăn xếp các lớp tích chập O(n/k) trong trường hợp các hạt nhân liền kề, hoặc O(logk(n)) trong trường hợp các tích chập giãn nở [18], làm tăng độ dài của các đường dẫn dài nhất giữa bất kỳ hai vị trí nào trong mạng. Các lớp tích chập thường đắt hơn các lớp hồi quy, theo hệ số k. Tuy nhiên, các tích chập tách rời [6] làm giảm đáng kể độ phức tạp, xuống còn O(k · n · d + n · d tích chập bằng sự kết hợp của một lớp tự chú ý và một lớp truyền thẳng từng điểm, ^{2}). Tuy nhiên, ngay cả với k = n, độ phức tạp của một cách tiếp cận mà chúng tôi áp dụng trong mô hình của mình.

Một lợi ích phụ khác là sự tự chú ý có thể tạo ra các mô hình dễ diễn giải hơn. Chúng tôi kiểm tra phân phối chú ý từ các mô hình của mình và trình bày cũng như thảo luận các ví dụ trong phần phụ lục. Không chỉ các đầu chú ý riêng lẻ học cách thực hiện các nhiệm vụ khác nhau một cách rõ ràng, nhiều đầu dường như còn thể hiện hành vi liên quan đến cấu trúc cú pháp và ngữ nghĩa của câu.

5 Đào tạo

Phần này mô tả chế độ đào tạo cho các mô hình của chúng tôi.

5.1 Dữ liệu đào tạo và phân lô

Chúng tôi đã đào tạo trên bộ dữ liệu WMT 2014 chuẩn Anh-Đức, bao gồm khoảng 4,5 triệu cặp câu. Các câu được mã hóa bằng phương pháp mã hóa cặp byte [3], với vốn từ vựng nguồn-đích chung khoảng 37.000 mã thông báo. Đối với tiếng Anh-Pháp, chúng tôi đã sử dụng bộ dữ liệu WMT 2014 Anh-Pháp lớn hơn đáng kể, bao gồm 36 triệu câu và các mã thông báo được chia thành vốn từ vựng 32.000 từ [38]. Các cặp câu được nhóm lại với nhau theo độ dài chuỗi gần đúng. Mỗi nhóm đào tạo chứa một tập hợp các cặp câu, bao gồm khoảng 25.000 mã thông báo nguồn và 25.000 mã thông báo đích.

5.2 Phần cứng và Lịch trình

Chúng tôi đã huấn luyện các mô hình trên một máy với 8 GPU NVIDIA P100. Đối với các mô hình cơ sở sử dụng các siêu tham số được mô tả trong toàn bộ bài báo, mỗi bước huấn luyện mất khoảng 0,4 giây. Chúng tôi đã huấn luyện các mô hình cơ sở với tổng cộng 100.000 bước, tương đương 12 giờ. Đối với các mô hình lớn (được mô tả ở dòng cuối cùng của bảng 3), thời gian bước huấn luyện là 1,0 giây. Các mô hình lớn được huấn luyện với 300.000 bước (3,5 ngày).

5.3 Trình tối ưu hóa

Chúng tôi đã sử dụng trình tối ưu hóa Adam [20] với β1 = 0,9, β2 = 0,98 và ε = 10 9 . Chúng tôi đã thay đổi tốc độ học trong suốt quá trình đào tạo, theo công thức:

$$\text{lrate} = d_{\text{người mẫu}}^{0,5} \cdot \min(\text{step\_num}^{0.5}, \text{step\_num} \cdot \text{warmup\_steps}^{1.5}) \quad (3)$$

Điều này tương ứng với việc tăng tốc độ học tuyến tính cho các bước huấn luyện warmup_steps đầu tiên , và sau đó giảm dần theo tỷ lệ nghịch với căn bậc hai của số bước. Chúng tôi sử dụng warmup_steps = 4000.

5.4 Chính quy hóa

Chúng tôi sử dụng ba loại chính quy hóa trong quá trình đào tạo:

Machine Translated by Google

Bảng 2: Máy biến áp đạt điểm BLEU tốt hơn so với các mô hình tiên tiến trước đây trên
Bài kiểm tra newstest2014 từ tiếng Anh sang tiếng Đức và từ tiếng Anh sang tiếng Pháp chỉ với một phần nhỏ chi phí đào tạo.

Người mẫu	MÀU XANH DA TRỜI		Chi phí đào tạo (FLOPs)	
	EN-DE	EN-FR	23.75 EN-DE	EN-FR
Mạng Byte [18]		39,2		1.0 · 1020
Deep-Att + PosUnk [39]	24,6	39,92	2.3 · 1019	1.4 · 1020
GNMT + RL [38]	25,16	40,46	9,6 · 1018	1,5 · 1020
ConvS2S [9]	26,03	40,56	2.0 · 1019	1.2 · 1020
Bộ Giáo dục và Đào tạo [32]		40,4		8.0 · 1020
Bộ đồng phục Deep-Att + PosUnk [39]	26.30	41,16	1,8 · 1020	1,1 · 1021
GNMT + RL Ensemble [38]	26.36	41,29	7,7 · 1019	1,2 · 1021
ConvS2S Ensemble [9]	27.3	38.1	3.3 · 1018	
Máy biến áp (mô hình cơ bản)	28,4	41,8	2.3 · 1019	
Máy biến áp (lớn)				

Bỏ qua còn lại Chúng tôi áp dụng bỏ qua [33] cho đầu ra của mỗi lớp con, trước khi nó được thêm vào đầu vào lớp con và được chuẩn hóa. Ngoài ra, chúng tôi áp dụng dropout cho tổng của các phép nhúng và mã hóa vị trí trong cả ngăn xếp mã hóa và giải mã. Đối với mô hình cơ sở, chúng tôi sử dụng tốc độ Pdrop = 0,1.

Làm mịn nhãn Trong quá trình đào tạo, chúng tôi đã sử dụng làm mịn nhãn có giá trị εls = 0,1 [36]. Điều này gây ra sự bối rối, vì mô hình học cách trở nên không chắc chắn hơn, nhưng cải thiện độ chính xác và điểm BLEU.

6 Kết quả

6.1 Dịch máy

Trong bài dịch tiếng Anh sang tiếng Đức của WMT 2014, mô hình máy biến áp lớn (Máy biến áp (lớn) trong Bảng 2) vượt trội hơn các mô hình tốt nhất đã được báo cáo trước đó (bao gồm cả các tập hợp) hơn 2,0 BLEU, thiết lập điểm BLEU hiện đại mới là 28,4. Cấu hình của mô hình này là được liệt kê ở dòng cuối cùng của Bảng 3. Việc đào tạo mất 3,5 ngày trên 8 GPU P100. Ngay cả mô hình cơ sở của chúng tôi vượt qua tất cả các mô hình và tập hợp đã công bố trước đó, với chi phí đào tạo chỉ bằng một phần nhỏ so với bất kỳ các mô hình cạnh tranh.

Trong nhiệm vụ dịch thuật từ tiếng Anh sang tiếng Pháp của WMT 2014, mô hình lớn của chúng tôi đạt được điểm BLEU là 41,0, vượt trội hơn tất cả các mô hình đơn lẻ đã công bố trước đó, với chi phí đào tạo ít hơn 1/4 mô hình tiên tiến trước đây. Mô hình Transformer (lớn) được đào tạo để dịch từ tiếng Anh sang tiếng Pháp đã được sử dụng tỷ lệ bỏ học Pdrop = 0,1, thay vì 0,3.

Đối với các mô hình cơ sở, chúng tôi đã sử dụng một mô hình duy nhất thu được bằng cách lấy trung bình 5 điểm kiểm tra cuối cùng, được viết cách nhau 10 phút. Đối với các mô hình lớn, chúng tôi đã tính trung bình 20 điểm kiểm tra cuối cùng. Chúng tôi đã sử dụng tìm kiếm chùm tia với kích thước chùm tia là 4 và hình phạt chiều dài α = 0,6 [38]. Các siêu tham số này đã được chọn sau khi thử nghiệm trên bộ phát triển. Chúng tôi thiết lập độ dài đầu ra tối đa trong suy luận về độ dài đầu vào + 50, nhưng kết thúc sớm khi có thể [38].

Bảng 2 tóm tắt kết quả của chúng tôi và so sánh chất lượng dịch thuật và chi phí đào tạo của chúng tôi với các mô hình khác kiến trúc từ tài liệu. Chúng tôi ước tính số lượng phép toán dấu chấm động được sử dụng để đào tạo một mô hình bằng cách nhân thời gian đào tạo, số lượng GPU được sử dụng và ước tính về thời gian duy trì khả năng xử lý dấu chấm động độ chính xác đơn của mỗi GPU [5].

6.2 Các biến thể mô hình

Để đánh giá tầm quan trọng của các thành phần khác nhau của Máy biến áp, chúng tôi đã thay đổi mô hình cơ sở của mình theo những cách khác nhau, đo lường sự thay đổi về hiệu suất trên bản dịch tiếng Anh sang tiếng Đức trên

[5]Chúng tôi sử dụng các giá trị lần lượt là 2,8, 3,7, 6,0 và 9,5 TFLOPS cho K80, K40, M40 và P100.

Machine Translated by Google

Bảng 3: Các biến thể trên kiến trúc Transformer. Các giá trị không được liệt kê giống hệt với các giá trị của cơ sở mô hình. Tất cả các số liệu đều nằm trong bộ phát triển dịch thuật từ tiếng Anh sang tiếng Đức, newstest2013. Đã liệt kê sự bối rối là theo từng từ, theo mã hóa cặp byte của chúng tôi, và không nên so sánh với sự bối rối theo từng từ.

	N dmodel	dff	h	dk	dv	Pdrop	εls	đào tạo bước	các tham số ×106	PPL (độ lệch)	BLEU (độ lệch)
cơ số 6	512	2048	8	64	64	0,1	0,1	100K	4,92	5,29	25,8	65
(MỘT)			1	512	512					5,00	24,9	
			4	128	128					4,91	25,5	
			16	32	32					5,01	25,8	
											25,4	
(B)				16						5,16	25,1	58
				32						5,01	25,4	60
(C)	2									6,11	23,7	36
	4									5,19	25,3	50
	8									4,88	25,5	80
		256		32	32					5,75	24,5	28
		1024		128						4,66	26,0	168
			1024							5,12	25,4	53
			4096							4,75	26,2	90
(Đ)						0,0				5,77	24,6	
						0,2				4,95	25,5	
							0,0			4,67	25.3	
							0,2			5,47	25,7	
(E) nhúng vị trí thay vì hình sin			lớn							4,92	25,7	
	6	1024	4096	16		0,3		300K	4,33		26,4	213

bộ phát triển, newstest2013. Chúng tôi đã sử dụng tìm kiếm chùm tia như mô tả trong phần trước, nhưng không trung bình điểm kiểm tra. Chúng tôi trình bày những kết quả này trong Bảng 3.

Trong Bảng 3 hàng (A), chúng tôi thay đổi số lượng tiêu đề chú ý và các chiều khóa chú ý và giá trị, giữ nguyên lượng tính toán không đổi, như được mô tả trong Phần 3.2.2. Trong khi đầu đơn chú ý là 0,9 BLEU tệ hơn cài đặt tốt nhất, chất lượng cũng giảm khi có quá nhiều đầu.

Trong Bảng 3 hàng (B), chúng tôi quan sát thấy việc giảm kích thước khóa chú ý dk làm giảm chất lượng mô hình. Điều này cho thấy rằng việc xác định khả năng tương thích không dễ dàng và khả năng tương thích phức tạp hơn Hàm số hơn tích vô hướng có thể có lợi. Chúng ta quan sát thêm ở hàng (C) và (D) rằng, đúng như dự đoán, Các mô hình lớn hơn thì tốt hơn, và dropout rất hữu ích trong việc tránh hiện tượng quá khớp. Ở hàng (E), chúng ta thay thế mã hóa vị trí hình sin với nhúng vị trí đã học [9] và quan sát gần như giống hệt nhau kết quả cho mô hình cơ sở.

6.3 Phân tích khu vực bầu cử tiếng Anh

Để đánh giá xem Transformer có thể khái quát hóa cho các nhiệm vụ khác hay không, chúng tôi đã thực hiện các thí nghiệm về tiếng Anh phân tích thành phần bầu cử. Nhiệm vụ này đặt ra những thách thức cụ thể: đầu ra phải tuân theo cấu trúc mạnh mẽ ràng buộc và dài hơn đáng kể so với đầu vào. Hơn nữa, chuỗi RNN-to-chuỗi các mô hình không thể đạt được kết quả tiên tiến nhất trong chế độ dữ liệu nhỏ [37].

Chúng tôi đã đào tạo một máy biến áp 4 lớp với dmodel = 1024 trên phần Tạp chí Phố Wall (WSJ) của Penn Treebank [25], khoảng 40 nghìn câu huấn luyện. Chúng tôi cũng huấn luyện nó trong một môi trường bán giám sát, sử dụng các tập đoàn dữ liệu BerkleyParser có độ tin cậy cao hơn với khoảng 17 triệu câu [37]. Chúng tôi đã sử dụng vốn từ vựng gồm 16K mã thông báo chỉ dành cho cài đặt WSJ và vốn từ vựng gồm 32K mã thông báo cho bối cảnh bán giám sát.

Chúng tôi chỉ thực hiện một số ít thí nghiệm để chọn ra những người bỏ cuộc, cả sự chú ý và sự còn lại (phần 5.4), tốc độ học tập và kích thước chùm tia trên bộ phát triển Phần 22, tất cả các tham số khác vẫn không thay đổi so với mô hình dịch thuật cơ sở từ tiếng Anh sang tiếng Đức. Trong quá trình suy luận, chúng tôi

Machine Translated by Google

Bảng 4: Bộ chuyển đổi tổng quát hóa tốt đối với việc phân tích thành phần tiếng Anh (Kết quả có ở Phần 23 (WSJ)

Trình phân tích cú pháp	Đào tạo	WSJ 23 F1
Vinyals & Kaiser el al. (2014) [37]	Chỉ WSJ, chỉ WSJ phân biệt, chỉ WSJ	88,3
Petrov và cộng sự (2006) [29]	phân biệt, chỉ WSJ phân	90,4
Zhu và cộng sự (2013) [40]	biệt, chỉ WSJ phân biệt,	90,4
Dyer và cộng sự (2016) [8]	phân biệt bán giám sát bán	91,7
Máy biến áp (4 lớp)	giám sát bán giám sát bán	91,3
Zhu và cộng sự (2013) [40]	giám sát bán giám	91,3
Hoàng và Harper (2009) [14]	sát đa nhiệm vụ	91,3
McClosky và cộng sự (2006) [26]	tạo ra	92,1
Vinyals & Kaiser và cộng sự. (2014) [37]		92,1
Máy biến áp (4 lớp)		92,7
Luong et al. (2015) [23]		93,0
Dyer và cộng sự (2016) [8]		93,3

tăng chiều dài đầu ra tối đa lên chiều dài đầu vào + 300. Chúng tôi sử dụng kích thước chùm tia là 21 và α = 0,3 chỉ dành cho WSJ và chế độ giám sát bán phần.

Kết quả của chúng tôi trong Bảng 4 cho thấy rằng mặc dù thiếu điều chỉnh cụ thể cho từng nhiệm vụ, mô hình của chúng tôi vẫn hoạt động tốt một cách đáng ngạc nhiên, mang lại kết quả tốt hơn tất cả các mô hình đã báo cáo trước đây ngoại trừ Ngữ pháp mạng nơ-ron hồi quy [8].

Ngược lại với các mô hình chuỗi-sang-chuỗi RNN [37], Transformer vượt trội hơn Berkeley- Parser [29] ngay cả khi chỉ đào tạo trên bộ đào tạo WSJ gồm 40K câu.

7 Kết luận

Trong công trình này, chúng tôi đã trình bày Transformer, mô hình chuyển đổi chuỗi đầu tiên hoàn toàn dựa trên chú ý, thay thế các lớp tuần hoàn thường được sử dụng nhất trong kiến trúc mã hóa-giải mã bằng sự chú ý nhiều chiều vào bản thân.

Đối với các tác vụ dịch thuật, Transformer có thể được đào tạo nhanh hơn đáng kể so với các kiến trúc dựa trên trên các lớp hồi quy hoặc lớp tích chập. Trên cả WMT 2014 từ tiếng Anh sang tiếng Đức và WMT 2014 Nhiệm vụ dịch thuật từ tiếng Anh sang tiếng Pháp, chúng tôi đạt đến trình độ tiên tiến mới. Trong nhiệm vụ trước, chúng tôi đã nỗ lực hết mình mô hình thậm chí còn vượt trội hơn tất cả các nhóm đã báo cáo trước đó.

Chúng tôi rất hào hứng về tương lai của các mô hình dựa trên sự chú ý và có kế hoạch áp dụng chúng vào các nhiệm vụ khác. Chúng tôi có kế hoạch mở rộng Transformer sang các vấn đề liên quan đến các phương thức đầu vào và đầu ra khác ngoài văn bản và để điều tra các cơ chế chú ý cục bộ, hạn chế để xử lý hiệu quả các đầu vào và đầu ra lớn chẳng hạn như hình ảnh, âm thanh và video. Giảm tính tuần tự của quá trình tạo ra sản phẩm là một mục tiêu nghiên cứu khác của chúng tôi.

Mã chúng tôi sử dụng để đào tạo và đánh giá các mô hình của mình có sẵn tại https://github.com/tensorflow/tensor2tensor.

Lời cảm ơn Chúng tôi biết ơn Nal Kalchbrenner và Stephan Gouws vì những đóng góp hữu ích của họ bình luận, chỉnh sửa và truyền cảm hứng.

Tài liệu tham khảo

[1] Jimmy Lei Ba, Jamie Ryan Kiros và Geoffrey E Hinton. Chuẩn hóa lớp. Bản in trước arXiv arXiv:1607.06450, 2016.

[2] Dzmitry Bahdanau, Kyunghyun Cho, và Yoshua Bengio. Dịch máy thần kinh được thực hiện bởi sự hợp tác học cách căn chỉnh và dịch chuyển. CoRR, abs/1409.0473, 2014.

[3] Denny Britz, Anna Goldie, Minh-Thang Luong, và Quoc V. Le. Khám phá sâu rộng về thần kinh Kiến trúc dịch máy. CoRR, abs/1703.03906, 2017.

[4] Jianpeng Cheng, Li Dong và Mirella Lapata. Mạng lưới bộ nhớ dài hạn ngắn cho máy đọc. Bản in trước arXiv arXiv:1601.06733, 2016.

Machine Translated by Google

[5] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk và Yoshua Bengio. Học biểu diễn cụm từ bằng bộ mã hóa-giải mã rnn cho dịch máy thống kê. CoRR, abs/1406.1078, 2014.

[6] Francois Chollet. Xception: Học sâu với phép tích chập tách biệt theo chiều sâu. arXiv bản in trước arXiv:1610.02357, 2016.

[7] Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho và Yoshua Bengio. Đánh giá thực nghiệm về mạng nơ-ron hồi quy có cổng trên mô hình chuỗi. CoRR, abs/1412.3555, 2014.

[8] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros và Noah A. Smith. Ngữ pháp mạng nơ-ron hồi quy. Trong Proc. của NAACL, 2016.

[9] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats và Yann N. Dauphin. Học trình tự tích chập sang trình tự. Bản in trước arXiv arXiv:1705.03122v2, 2017.

[10] Alex Graves. Tạo chuỗi với mạng nơ-ron hồi quy. Bản in trước arXiv arXiv:1308.0850, 2013.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren và Jian Sun. Học dư sâu để nhận dạng hình ảnh. Trong Kỷ yếu Hội nghị IEEE về Thị giác máy tính và Nhận dạng mẫu, trang 770–778, 2016.

[12] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi và Jürgen Schmidhuber. Dòng chuyển màu trong lưới hồi quy: khó khăn trong việc học các mối quan hệ phụ thuộc dài hạn, 2001.

[13] Sepp Hochreiter và Jürgen Schmidhuber. Trí nhớ ngắn hạn dài. Tính toán thần kinh, 9(8):1735–1780, 1997.

[14] Zhongqiang Huang và Mary Harper. Ngữ pháp PCFG tự huấn luyện với chú thích tiềm ẩn trên nhiều ngôn ngữ. Trong Kỷ yếu Hội nghị năm 2009 về Phương pháp Thực nghiệm trong Xử lý Ngôn ngữ Tự nhiên, trang 832–841. ACL, tháng 8 năm 2009.

[15] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer và Yonghui Wu. Khám phá giới hạn của mô hình hóa ngôn ngữ. Bản in trước arXiv arXiv:1602.02410, 2016.

[16] Łukasz Kaiser và Samy Bengio. Trí nhớ chủ động có thể thay thế sự chú ý không? Trong Những tiến bộ trong thần kinh Hệ thống xử lý thông tin (NIPS), 2016.

[17] Łukasz Kaiser và Ilya Sutskever. GPU thần kinh học các thuật toán. Trong Hội nghị quốc tế về biểu diễn học tập (ICLR), 2016.

[18] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves và Ko-ray Kavukcuoglu. Dịch máy thần kinh trong thời gian tuyến tính. Bản in trước arXiv arXiv:1610.10099v2, 2017.

[19] Yoon Kim, Carl Denton, Luong Hoang và Alexander M. Rush. Mạng lưới chú ý có cấu trúc. Trong Hội nghị quốc tế về biểu diễn học tập, 2017.

[20] Diederik Kingma và Jimmy Ba. Adam: Một phương pháp tối ưu hóa ngẫu nhiên. Trong ICLR, 2015.

[21] Oleksii Kuchaiev và Boris Ginsburg. Các thủ thuật phân tích nhân tử cho mạng LSTM. Bản in trước arXiv arXiv:1703.10722, 2017.

[22] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou và Yoshua Bengio. Nhúng câu tự chú ý có cấu trúc. Bản thảo trước arXiv arXiv:1703.03130, 2017.

[23] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals và Lukasz Kaiser. Học trình tự đa nhiệm vụ sang trình tự. Bản thảo arXiv arXiv:1511.06114, 2015.

[24] Minh-Thang Luong, Hieu Pham, và Christopher D Manning. Các phương pháp tiếp cận hiệu quả đối với dịch máy thần kinh dựa trên sự chú ý. Bản thảo arXiv arXiv:1508.04025, 2015.

Machine Translated by Google

[25] Mitchell P Marcus, Mary Ann Marcinkiewicz và Beatrice Santorini. Xây dựng một kho ngữ liệu tiếng Anh có chú thích lớn: Ngân hàng cây Penn. Ngôn ngữ học tính toán, 19(2):313–330, 1993.

[26] David McClosky, Eugene Charniak và Mark Johnson. Tự đào tạo hiệu quả để phân tích cú pháp. Trong Biên bản Hội nghị Công nghệ Ngôn ngữ Con người của NAACL, Hội nghị Chính, trang 152–159. ACL, tháng 6 năm 2006.

[27] Ankur Parikh, Oscar Täckström, Dipanjan Das và Jakob Uszkoreit. Một mô hình chú ý có thể phân tích. Trong Phương pháp Thực nghiệm trong Xử lý Ngôn ngữ Tự nhiên, 2016.

[28] Romain Paulus, Caiming Xiong và Richard Socher. Một mô hình được củng cố sâu sắc cho trừu tượng tóm tắt. Bản in trước arXiv arXiv:1705.04304, 2017.

[29] Slav Petrov, Leon Barrett, Romain Thibaux và Dan Klein. Học chú thích cây chính xác, gọn nhẹ và dễ diễn giải. Trong Kỷ yếu Hội nghị Quốc tế lần thứ 21 về Ngôn ngữ học Tính toán và Hội nghị thường niên lần thứ 44 của ACL, trang 433–440. ACL, tháng 7 năm 2006.

[30] Ofir Press và Lior Wolf. Sử dụng nhúng đầu ra để cải thiện các mô hình ngôn ngữ. arXiv bản in trước arXiv:1608.05859, 2016.

[31] Rico Sennrich, Barry Haddow và Alexandra Birch. Bản dịch máy thần kinh của các từ hiếm có đơn vị từ phụ. Bản in trước arXiv arXiv:1508.07909, 2015.

[32] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton và Jeff Dean. Mạng nơ-ron cực lớn: Lớp hỗn hợp chuyên gia có cổng thưa thớt . Bản in trước arXiv arXiv:1701.06538, 2017.

[33] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever và Ruslan Salakhutdi- nov. Dropout: một cách đơn giản để ngăn chặn mạng nơ-ron khỏi quá khớp. Tạp chí Nghiên cứu Học máy , 15(1):1929–1958, 2014.

[34] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston và Rob Fergus. Mạng bộ nhớ đầu cuối. Trong C. Cortes, ND Lawrence, DD Lee, M. Sugiyama và R. Garnett, biên tập viên, Những tiến bộ trong Hệ thống xử lý thông tin thần kinh 28, trang 2440–2448. Curran Associates, Inc., 2015.

[35] Ilya Sutskever, Oriol Vinyals và Quoc VV Le. Học trình tự với mạng nơ-ron. Trong Những tiến bộ trong Hệ thống xử lý thông tin nơ-ron, trang 3104–3112, 2014.

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, và Zbigniew Wojna. Xem xét lại kiến trúc khởi đầu cho thị giác máy tính. CoRR, abs/1512.00567, 2015.

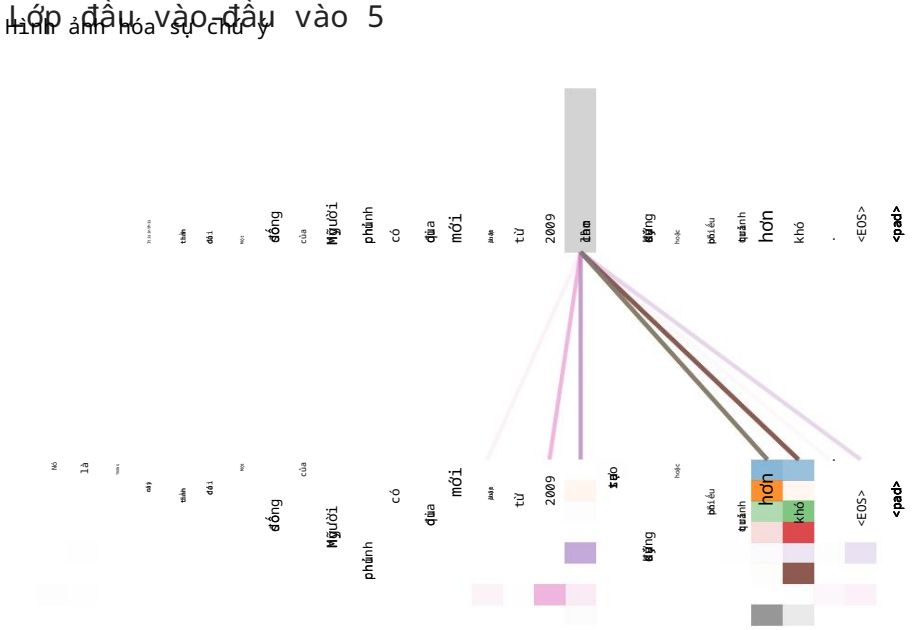
[37] Vinyals & Kaiser, Koo, Petrov, Sutskever và Hinton. Ngữ pháp như một ngôn ngữ nước ngoài. Trong Những tiến bộ trong Hệ thống xử lý thông tin thần kinh, 2015.

[38] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Hệ thống dịch máy thần kinh của Google: Thu hẹp khoảng cách giữa bản dịch của con người và bản dịch máy. Bản in trước arXiv arXiv:1609.08144, 2016.

[39] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li và Wei Xu. Mô hình hồi quy sâu với kết nối chuyển tiếp nhanh cho dịch máy thần kinh. CoRR, abs/1606.04199, 2016.

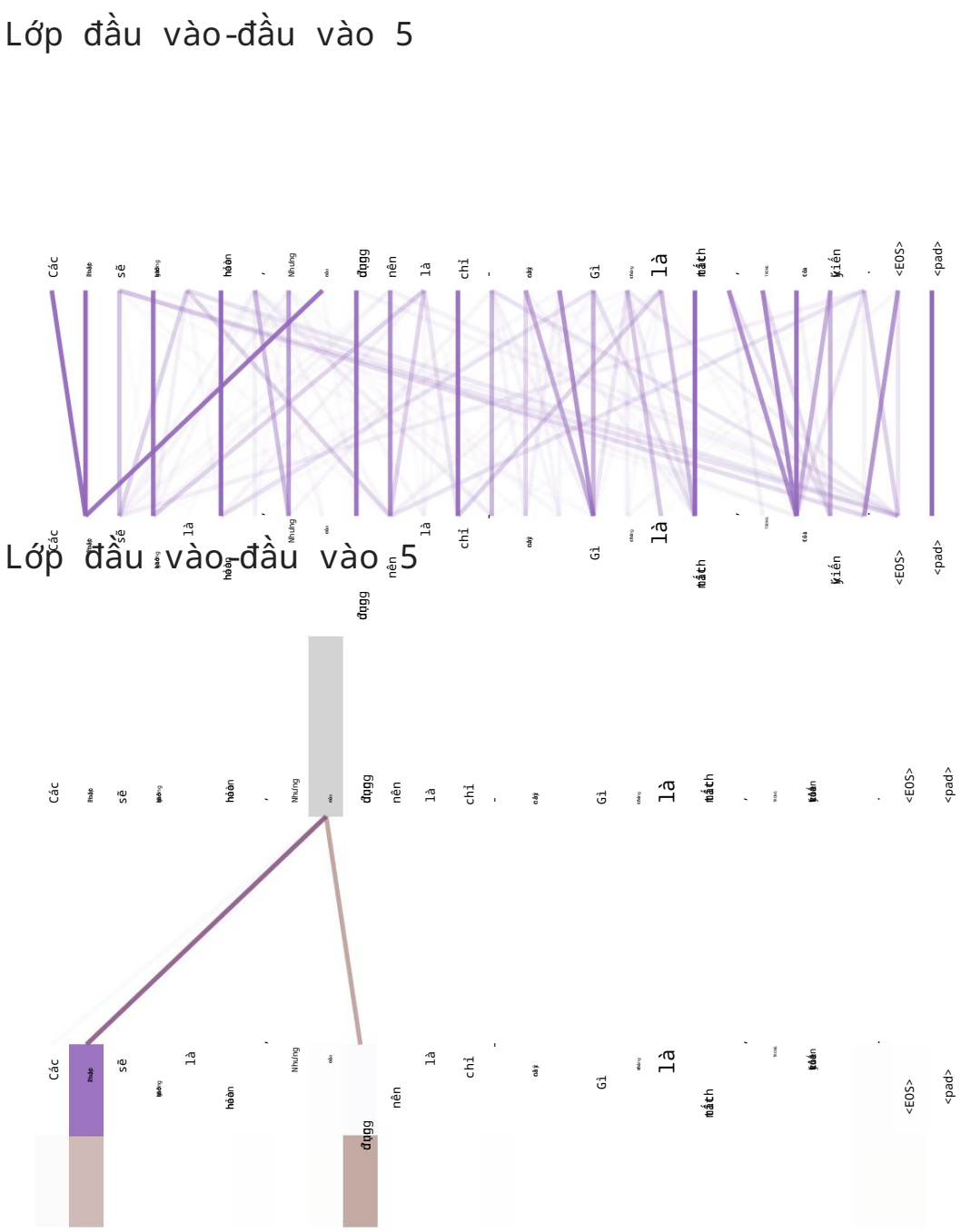
[40] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang và Jingbo Zhu. Phân tích cú pháp thành phần dịch chuyển-giảm nhanh và chính xác. Trong Kỷ yếu Hội nghị thường niên lần thứ 51 của ACL (Tập 1: Bài báo dài), trang 434–443. ACL, tháng 8 năm 2013.

Machine Translated by Google



Hình 3: Một ví dụ về cơ chế chú ý theo các phụ thuộc đường dài trong sự tự chú ý của bộ mã hóa ở lớp 5/6. Nhiều đầu chú ý chú ý đến sự phụ thuộc xa của động từ 'making', hoàn thành cụm từ 'making...more difficult'. Các chú ý ở đây chỉ hiển thị cho từ 'making'. Các màu khác nhau biểu thị các đầu khác nhau. Xem tốt nhất ở chế độ màu.

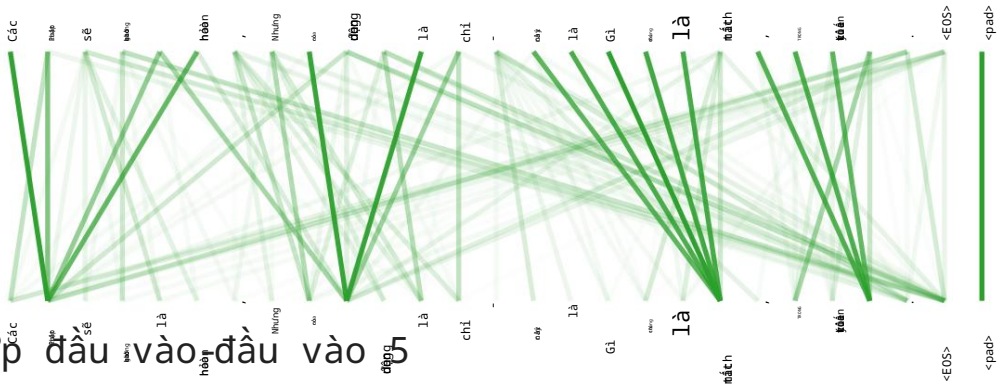
Machine Translated by Google



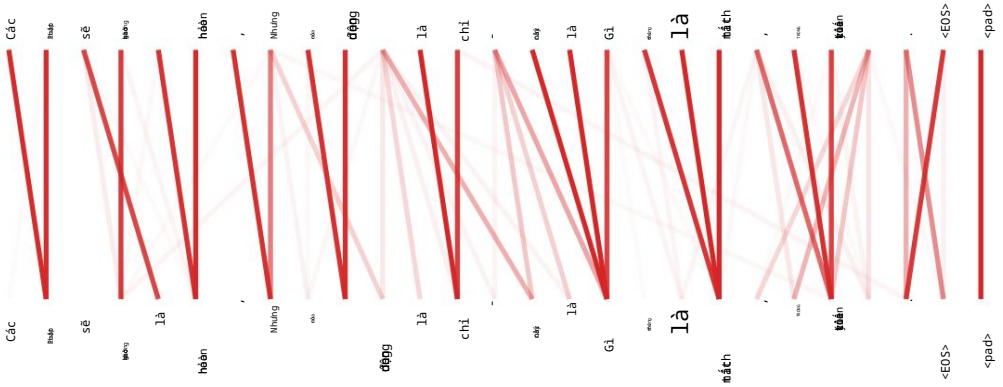
Hình 4: Hai đầu chú ý, cũng ở lớp 5 trong số 6, dường như có liên quan đến việc giải quyết phép ẩn dụ. Trên: Chú ý đầy đủ cho đầu 5. Dưới: Chú ý riêng biệt từ từ "its" cho đầu 5 và 6. Lưu ý rằng các chú ý rất sắc nét đối với từ này.

Machine Translated by Google

Lớp đầu vào-đầu vào 5



Lớp đầu vào-đầu vào 5



Hình 5: Nhiều đầu chú ý thể hiện hành vi dường như liên quan đến cấu trúc câu. Chúng tôi đưa ra hai ví dụ như vậy ở trên, từ hai đầu khác nhau của bộ mã hóa tự chú ý ở lớp 5/6. Rõ ràng là các đầu đã học cách thực hiện các nhiệm vụ khác nhau.