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
Google Research
nikip@google.com

Jakob Uszkoreit
Google Research
usz@google.com

Llion Jones
Google Research
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

trừu tượng

Các mô hình tải nạp trình tự ưu thế dựa trên các mạng thần kinh xoắn hoặc hồi quy phức tạp bao gồm bộ mã hóa và bộ giải mã. Các mô hình hoạt động tốt nhất cũng kết nối bộ mã hóa và bộ giải mã thông qua 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 khả năng lặp lại và tích chập. Các thử nghiệm trên hai tác vụ dịch máy cho thấy các mô hình này có chất lượng vượt trội trong khi khả năng song song hóa cao hơn và cần ít thời gian đào tạo hơn đáng kể. Mô hình của chúng tôi đạt được 28,4 BLEU trong nhiệm vụ dịch từ tiếng Anh sang tiếng Đức của WMT 2014, cải thiện hơn 2 BLEU so với các kết quả tốt nhất hiện có, bao gồm cả các bản hòa tấu. Trong nhiệm vụ dịch từ tiếng Anh sang tiếng Pháp của WMT 2014, mô hình của chúng tôi thiết lập điểm số BLEU hiện đại nhất cho một mô hình mới là 41,8 sau khi đào tạo trong 3,5 ngày trên tám GPU, một phần nhỏ chi phí đào tạo tốt nhất mô hình từ văn học. Chúng tôi cho thấy rằng Transformer tổng quát hóa tốt cho các nhiệm vụ khác bằng cách áp dụng thành công nó vào phân tích cú pháp khu vực bầu cử bằng tiếng Anh cả với dữ liệu đào tạo lớn và hạn chế.

1. Giới thiệu

Mạng thần kinh hồi quy, bộ nhớ ngắn hạn dài [13] và đặc biệt là mạng thần kinh tái phát có kiểm soát [7] , đã được thiết lập vững chắc như là phương pháp tiếp cận hiện đại trong mô hình hóa trình tự và

Đóng góp bình đẳng. Thứ tự danh sách là ngẫu nhiên. Jakob đã đề xuất thay thế RNN bằng tính năng tự chú ý 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 chủ yếu vào mọi khía cạnh của công việc này. Noam đã đề xuất sự chú ý của sản phẩm chấm được chia tỷ lệ, sự chú ý của nhiều đầu và biểu diễn vị trí không có tham số và trở thành người khác 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 các biến thể mô hình mới, chịu trách nhiệm về cơ sở mã ban đầu của chúng tôi cũng như khả năng suy luận và trực quan hóa hiệu quả. Lukasz và Aidan đã dành vô số ngày dài để thiết kế các phần khác nhau và triển khai tensor2tensor, thay thế cơ sở mã trước đây của chúng tôi, cải thiện đáng kể kết quả và đẩy nhanh quá trình nghiên cứu của chúng tôi.

†Công việc được thực hiện khi ở Google Brain.

‡Công việc được thực hiện khi ở Google Research.

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

các vấn đề chuyển đổi như mô hình hóa ngôn ngữ và dịch máy [35, 2, 5]. Kể từ đó, nhiều nỗ lực đã tiếp tục mở rộng ranh giới của các mô hình ngôn ngữ lặp lại và kiến trúc bộ mã hóa-giải mã [38, 24, 15].

Các mô hình hồi quy thư ờng tính toán nhân tố dọc theo các vị trí ký hiệu của chuỗi đầu vào và đầu ra. Sắp xếp các vị trí theo các bư ớc trong thời gian tính toán, chúng tạo ra một chuỗi các trạng thái ẩn ht, như một hàm của trạng thái ẩn ht 1 trư ớc đó và đầu vào cho vị trí t. Bản chất tuần tự vốn có này ngăn cản quá trình song song hóa trong các ví dụ đào tạo, điều này trở nên quan trọng ở độ dài chuỗi dài hơn, vì các ràng buộc bộ nhớ giới hạn việc xử lý theo đợt giữa các ví dụ. Công việc gần đây đã đạt đư ợc những cải tiến đáng kể về hiệu quả tính toán thông qua các thủ thuật phân tích thừa số [21] và tính toán có điều kiện [32], đồng thời cải thiện hiệu suất của mô hình trong trư ờng hợp sau. Tuy nhiên, hạn chế cơ bản của tính toán tuần tự vẫn còn.

Các cơ chế chú ý đã trở thành một phần không thể thiếu của các mô hình truyền tải và mô hình trình tự hấp dẫn trong các tác vụ khác nhau, cho phép mô hình hóa các phụ thuộc mà không cần quan tâm đến khoảng cách của chúng trong các trình tự đầu vào hoặc đầu ra [2, 19]. Tuy nhiên, trong tất cả trừ một vài trư ờng hợp [27], các cơ chế chú ý như vậy đư ợc sử dụng cùng với mạng hồi quy.

Trong công việc này, chúng tôi đề xuất Máy biến áp, một kiến trúc mô hình tránh sự lặp lại và thay vào đó hoàn toàn dựa vào cơ chế chú ý để thu hút sự phụ thuộc toàn cầu giữa đầu vào và đầu ra. Transformer cho phép khả năng song song hóa cao hơn đáng kể và có thể đạt đến trình độ mới về chất lư ợng bản dịch sau khi đư ợc đào tạo trong ít nhất mư ời hai giờ trên tám GPU P100.

2 bối cảnh

Mục tiêu giảm tính toán tuần tự cũng tạo thành nền tảng của GPU thần kinh mở rộng [16], ByteNet [18] và ConvS2S [9], tất cả đều sử dụng mạng thần kinh tích chập làm khối xây dựng cơ bản, tính toán song song các biểu diễn ẩn cho tất cả đầu vào và các vị trí đầu ra. Trong các mô hình này, số lư ợng thao tác cần thiết để liên kết 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 theo logarit đối với ByteNet. Điều này làm cho việc tìm hiểu sự phụ thuộc giữa các vị trí ở xa trở nên khó khăn hơn [12]. Trong Máy biến áp, điều này đư ợc giảm xuống một số lư ợng hoạt động không đổi, mặc dù phải trả giá bằng việc giảm độ phân giải hiệu quả do lấy trung bình các vị trí có trọng số chú ý, một hiệu ứng mà chúng tôi khắc phục bằng Chú ý nhiều đầu như đư ợc mô tả trong phần 3.2 .

Tự chú ý, đôi khi đư ợc gọi là chú ý nội bộ là một cơ chế chú ý liên quan đến các vị trí khác nhau của một chuỗi đơn lẻ để tính toán biểu diễn của chuỗi. Tự chú ý đã đư ợc sử dụng thành công trong nhiều nhiệm vụ khác nhau bao gồm đọc hiểu, tóm tắt trừu tư ợng, dẫn xuất văn bản và biểu diễn câu độc lập với nhiệm vụ học tập [4, 27, 28, 22].

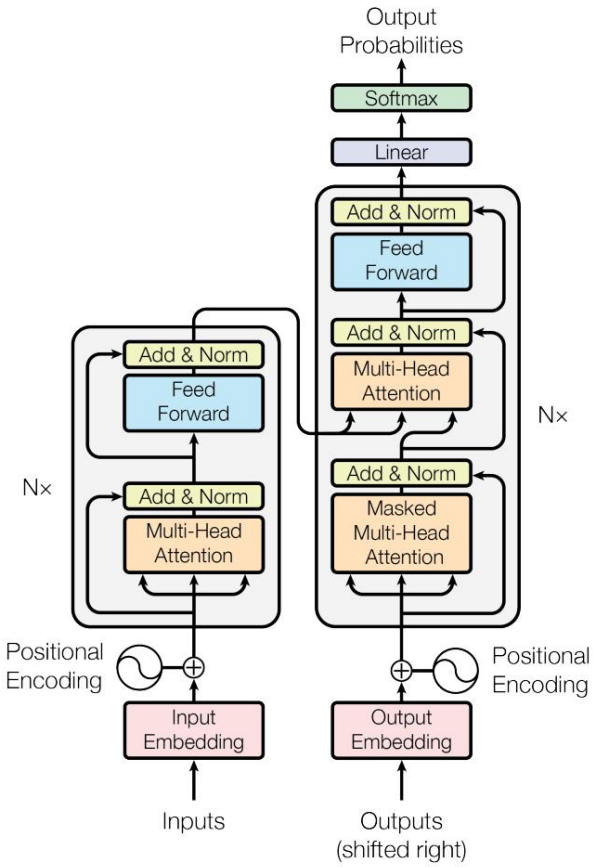
Các mạng bộ nhớ end-to-end dựa trên cơ chế chú ý lặp lại thay vì lặp lại theo trình tự và đã đư ợc chứng minh là hoạt động tốt trên các nhiệm vụ mô hình hóa ngôn ngữ và trả lời câu hỏi bằng ngôn ngữ đơn giản [34].

Tuy nhiên, theo hiểu biết tốt nhất của chúng tôi, Transformer là mô hình tải nạp đầu tiên hoàn toàn dựa vào khả năng tự chú ý để tính toán các biểu diễn đầu vào và đầu ra của nó mà không sử dụng tích chập hoặc RNN đư ợc căn chỉnh theo trình tự. Trong các phần tiếp theo, chúng tôi sẽ mô tả Máy biến áp, thúc đẩy sự chú ý của bản thân và thảo luận về những ư u điểm của nó so với các mô hình như [17, 18] và [9].

3 Kiến trúc mô hình

Hầu hết các mô hình tải nạp chuỗi thần kinh cạnh tranh đều có cấu trúc bộ mã hóa-giải mã [5, 2, 35]. Ở đây, bộ mã hóa ánh xạ chuỗi biểu diễn ký hiệu đầu vào (x1, ..., xn) thành chuỗi biểu diễn liên tục z = (z1, ..., zn). Cho trư ớc 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 bư ớc, mô hình tự động hồi quy [10], sử dụng các ký hiệu đư ợc tạo trư ớc đó làm đầu vào bổ sung khi tạo bư ớc tiếp theo.

Máy biến áp tuân theo kiến trúc tổng thể này bằng cách sử dụng các lớp tự chú ý và theo điểm xếp chồng lên nhau, đư ợc kết nối đầy đủ cho cả bộ mã hóa và bộ giải mã, tư ơng ứng đư ợc hiển thị ở nửa bên trái và bên phải của Hình 1.



Hình 1: Kiến trúc mô hình Transformer.

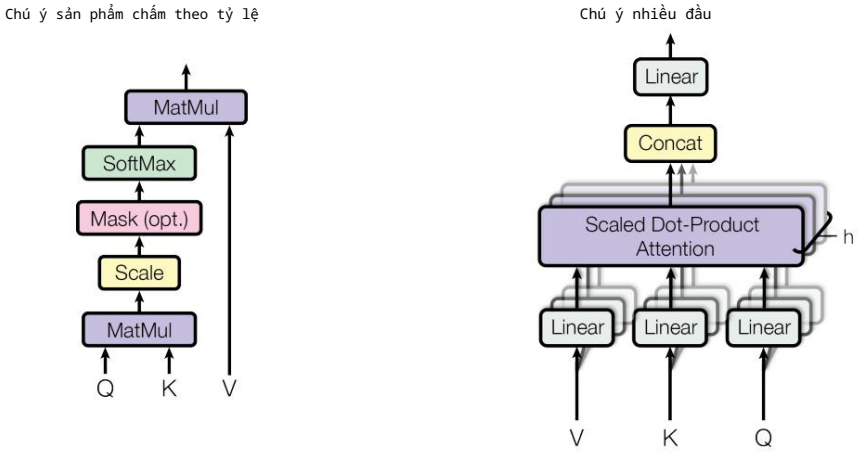
3.1 Ngăn xếp bộ mã hóa và giải mã

Bộ mã hóa: Bộ mã hóa bao gồm một chồng N = 6 lớp giống hệt nhau. Mỗi lớp có hai lớp con. Đầu tiên là cơ chế tự chú ý nhiều đầu và thứ hai là mạng chuyển tiếp nguồn cấp dữ liệu đơ ợc kết nối đầy đủ theo vị trí, đơn giản. Chúng tôi sử dụng kết nối còn lại [11] xung quanh mỗi lớp trong số hai lớp con, tiếp theo là chuẩn hóa lớp [1]. Nghĩa là, đầu ra của mỗi lớp con là LayerNorm(x + Lớp con(x)), trong đó Lớp con(x) là chức năng do chính lớp con đó thực hiện . Để tạo điều kiện thuận lợi cho các kết nối còn lại này, tất cả các lớp con trong mô hình, cũng như các lớp nhúng, tạo ra đầu ra có kích thư ớc 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 bộ mã hóa, bộ giải mã sẽ chèn một lớp con thứ ba, lớp này thực hiện chú ý nhiều đầu đối với đầu ra của ngăn xếp bộ mã hóa. Tư ơng tự như bộ mã hóa, chúng tôi sử dụng các kết nối còn lại xung quanh mỗi lớp phụ, sau đó là chuẩn hóa lớp. Chúng tôi cũng sửa đổi lớp con tự chú ý trong ngăn xếp bộ giải mã để ngăn các vị trí tham gia vào các vị trí tiếp theo. Mặt nạ này , kết hợp với thực tế là các phần nhúng đầu ra đư ợc bù bởi 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 đư ợc mô tả là ánh xạ một truy vấn và một tập hợp các cặp khóa-giá trị thành một đầu ra, trong đó truy vấn, khóa, giá trị và đầu ra đều là các vectơ. Đầu ra đư ợc tính toán dư ới dạng tổng trọng số của các giá trị, trong đó trọng số đư ợc gán cho từng giá trị đư ợc tính toán bằng hàm tư ơng thích của truy vấn với khóa tư ơng ứng.



Hình 2: (trái) Sự chú ý của sản phẩm chấm được chia tỷ lệ. (phải) Chú ý nhiều đầu bao gồm một số lớp chú ý chạy song song.

3.2.1 Chú ý sản phẩm chấm theo tỷ lệ

Chúng tôi gọi sự chú ý đặc biệt của mình là "Sự chú ý theo sản phẩm theo tỷ lệ" (Hình 2). Đầu vào bao gồm các truy vấn và khóa của thứ nguyên dk và các giá trị của thứ nguyên dv. Chúng tôi tính toán tích vô hướng của truy vấn với tất cả các khóa, chia từng 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 đồng thời trên một tập hợp các truy vấn, được đóng gói cùng nhau thành ma trận Q. Các khóa và giá trị cũng được đóng gói cùng nhau thành ma trận K và V . Chúng tôi tính toán ma trận đầu ra là:

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

Hai chức năng chú ý được sử dụng phổ biến nhất là chú ý cộng [2] và chú ý tích vô cực (đa bội). Sự chú ý của sản phẩm chấm giống với thuật toán của chúng tôi, ngoại trừ hệ số tỷ lệ của $\frac{1}{\sqrt{dk}}$. Chú ý bổ sung tính toán chức năng tương thích bằng cách sử dụng mạng chuyển tiếp nguồn cấp dữ liệu với một lớp ẩn duy nhất. Mặc dù cả hai đều giống nhau về độ phức tạp về mặt lý thuyết, nhưng trên thực tế, sự chú ý của sản phẩm chấm nhanh hơn và tiết kiệm không gian hơn nhiều, vì nó có thể được triển khai bằng cách sử dụng 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, thì tính chú ý bổ sung sẽ vượt trội hơn tính năng chú ý của sản phẩm chấm mà không cần chia tỷ lệ cho 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 lớn về độ lớn, đẩy hàm softmax vào các vùng có độ dốc cực nhỏ [4]. Để chống lại hiệu ứng này, chúng tôi chia tỷ lệ các tích vô hướng theo $\frac{1}{\sqrt{dk}}$.

3.2.2 Chú ý nhiều đầu

Thay vì thực hiện một chức năng chú ý duy nhất với các khóa, giá trị và truy vấn chiều dmodel, chúng tôi nhận thấy việc chiếu tuyến tính các truy vấn, khóa và giá trị h lần với các phép chiếu tuyến tính đã học khác nhau tương ứng với các chiều dk, dk và dv . Trên mỗi phiên bản dự kiến này của truy vấn, khóa và giá trị, sau đó chúng tôi thực hiện chức năng chú ý song song, mang lại giá trị đầu ra theo chiều dv . Chúng được nối và một lần nữa được chiếu, dẫn đến các giá trị cuối cùng, như được mô tả trong Hình 2.

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

Chú ý nhiều đầu cho phép mô hình cùng tham gia vào thông tin từ các không gian con biểu diễn khác nhau ở các vị trí khác nhau. Với một đầu chú ý duy nhất, tính trung bình sẽ hạn chế điều này.

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

$$\text{head}_i = \text{Chú ý}(QW^Q_i, KW^K_i, VW^V_i)$$

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

Trong công việc này, chúng tôi sử dụng h = 8 lớp chú ý song song hoặc các đầu. Đối với mỗi trong số này, chúng tôi sử dụng dk = dv = dmodel/h = 64. Do kích thước của mỗi đầu giảm, tổng chi phí tính toán tương tự như chi phí tính toán của một đầu với toàn bộ chiều.

3.2.3 Ứng dụng của 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 bộ giải mã-bộ giải mã", các truy vấn đến từ lớp bộ giải mã trước đó và 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 tham gia vào tất cả các vị trí trong chuỗi đầu vào. Điều này bắt chước các cơ chế chú ý của bộ mã hóa-giải mã điển hình trong các mô hình theo trình tự như [38, 2, 9].
- Bộ mã hóa chứa các lớp tự chú ý. Trong lớp tự chú ý, tất cả các khóa, giá trị và truy vấn đến từ cùng một nơi, trong trường hợp này là đầu ra của lớp trước trong bộ mã hóa. Mỗi vị trí trong bộ mã hóa có thể tham gia vào tất cả các vị trí trong lớp trước của bộ mã hóa.
- Tương tự, các lớp tự chú ý trong bộ giải mã cho phép mỗi vị trí trong bộ giải mã tham gia vào tất cả các vị trí trong bộ giải mã cho đến và bao gồm cả vị trí đó. Chúng ta cần ngăn luồng thông tin đi bên trái trong bộ giải mã để duy trì thuộc tính tự động hồi quy. Chúng tôi triển khai điều này bên trong sự chú ý của sản phẩm chấm được chia tỷ lệ bằng cách che dấu (đặt thành $-\infty$) 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 chuyển tiếp nguồn cấp dữ liệu theo vị trí

Ngoài các lớp phụ chú ý, mỗi lớp trong bộ mã hóa và bộ giải mã của chúng tôi chứa một mạng chuyển tiếp nguồn cấp dữ liệu được kết nối đầy đủ, được áp dụng cho từng vị trí riêng biệt và giống hệt nhau. Điều này bao gồm hai phép biến đổi tuyến tính với 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 trên các vị trí khác nhau, nhưng chúng sử dụng các tham số khác nhau từ lớp này sang lớp khác. Một cách khác để mô tả điều này là hai 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 tải nạp trình tự khác, chúng tôi sử dụng các phép nhúng đã học để chuyển đổi mã thông báo đầu vào và mã thông báo đầu ra thành vectơ của mô hình thứ nguyên. Chúng tôi cũng sử dụng phép chuyển đổi tuyến tính đã học thông thường và hàm softmax để chuyển đổi đầu ra của bộ giải mã thành xác suất mã thông báo 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 trước softmax, tương tự như [30]. Trong các lớp nhúng, chúng tôi nhân các trọng số đó với $\sqrt{\text{dmodel}}$.

3.5 Mã hóa vị trí

Vì mô hình của chúng tôi không chứa lặp lại và không tích chập, nên để mô hình sử dụng thứ tự của chuỗi, chúng tôi phải thêm một số thông tin về vị trí tương đối hoặc tuyệt đối của

Bảng 1: Độ dài đường dẫn tối đa, độ phức tạp trên mỗi lớp và số hoạt động tuần tự tối thiểu cho các loại lớp khác nhau. n là độ dài chuỗi, d là kích thước biểu diễn, k là kích thước hạt nhân của các kết cấu và r là kích thước của vùng lân cận trong sự tự chú ý bị hạn chế.

Loại lớp	Độ phức tạp trên mỗi lớp	Chiều dài đường dẫn tối đa tuần tự hoạt động	
tự chú ý	TRÊ N ^{2} ·	Ô(1)	Ô(1)
định kỳ	O(n · d ^{2} d))	TRÊ N)	TRÊ N)
tích chập	O(k · n · d ^{2})	Ô(1)	O(logk(n))
Tự quan tâm (hạn chế)	O(r · n · d)	Ô(1)	O(n/r)

mã thông báo trong chuỗi. Để làm được điều này, chúng tôi thêm "mã hóa theo vị trí" vào phần nhúng đầu vào ở cuối ngăn xếp bộ mã hóa và bộ giải mã. Các mã hóa vị trí có cùng kích thước dmodel như các nhúng, do đó có thể cộng lại cả hai. Có nhiều lựa chọn mã hóa vị trí, đã học và cố định [9].

Trong công việc này, chúng tôi sử dụng các hàm sin và cosin của các 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 kích thước. 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 tôi chọn chức năng này vì chúng tôi đưa ra giả thuyết rằng nó sẽ cho phép mô hình dễ dàng học cách tham dự theo các vị trí tương đối, vì đối với bất kỳ độ lệch k cố định nào , P Epos+k có thể được biểu diễn dưới dạng hàm tuyến tính của P Epos.

Thay vào đó , chúng tôi cũng đã thử nghiệm bằng cách sử dụng các nhúng vị trí đã học [9] và nhận thấy rằng hai phiên bản tạo ra kết quả gần như giống hệt nhau (xem hàng trong Bảng 3 (E)). Chúng tôi chọn phiên bản hình sin vì nó có thể cho phép mô hình ngoại suy theo độ dài trình tự dài hơn so với phiên bản gặp phải trong quá trình đào tạo.

4 Tại sao phải quan tâm đến bản thân

Trong phần này, chúng ta so sánh các khía cạnh khác nhau của các lớp tự chú ý với các lớp hồi quy và lớp chập thường được sử dụng để ánh xạ một chuỗi biểu diễn ký hiệu có độ dài thay đổi (x1, ..., xn) với một chuỗi khác có độ dài bằng nhau (z1, . .., zn), với xi chẳng hạn như, zlớp Pần trong bộ mã hóa hoặc bộ giải mã tải nạp trình tự điển hình. Thúc đẩy việc sử dụng sự chú ý đến bản thân, chúng tôi xem xét ba mong muốn. ^{d}

Một là tổng độ phức tạp tính toán trên mỗi lớp. Một cái khác là số lượng tính toán có thể được song song hóa, được đo bằng số lượng hoạt động tuần tự tối thiểu được yêu cầu.

Thứ ba là độ dài đường dẫn giữa các phụ thuộc tầm xa trong mạng. Học các phụ thuộc tầm xa là một thách thức chính trong nhiều nhiệm vụ tải nạp trình tự. Một yếu tố chính ảnh hưởng đến khả năng học các phụ thuộc như vậy là độ dài của các đường dẫn tín hiệu tiến và lùi phải đi qua mạng. Các đường dẫn này giữa bất kỳ sự kết hợp các vị trí nào trong chuỗi đầu vào và đầu ra càng ngắn thì càng dễ học các phụ thuộc tầm xa [12]. Do đó, chúng tôi cũng so sánh độ dài đường dẫn tối đa giữa hai vị trí đầu vào và đầu ra bất kỳ trong các mạng bao gồm các loại lớp khác nhau.

Như đã lưu ý trong Bảng 1, lớp tự chú ý kết nối tất cả các vị trí với số lượng hoạt động được thực hiện tuần tự không đổi, trong khi lớp lặp lại yêu cầu O(n) hoạt động tuần tự. Xét về độ phức tạp tính toán, các lớp tự chú ý nhanh hơn các lớp lặp lại khi độ dài chuỗi n nhỏ hơn chiều biểu diễn d, điều này thường xảy ra nhất với các biểu diễn câu được sử dụng bởi các mô hình hiện đại trong bản dịch máy , chẳng hạn như biểu diễn từ mảnh [38] và cặp byte [31] . Để cải thiện hiệu suất tính toán cho các tác vụ liên quan đến các chuỗi rất dài, có thể hạn chế việc tự chú ý đến việc chỉ xem xét một vùng lân cận có kích thước r trong

chuỗi đầu vào xoay 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 điều tra phương pháp này hơn nữa trong công việc trong tương lai.

Một lớp tích chập đơn với độ rộng nhân k < n không kết nối tất cả các cặp vị trí đầu vào và đầu ra. Làm như vậy yêu cầu một chồng các lớp chập O(n/k) trong trường hợp các nhân liền kề hoặc O(logk(n)) trong trường hợp các chập giãn nở [18], làm tăng độ dài của các đường dẫn dài nhất giữa hai vị trí bất kỳ trong mạng. Các lớp tích chập thường đắt hơn các lớp tái phát, theo hệ số k. Tuy nhiên, các tích chập có thể tách rời [6] làm giảm đáng kể độ phức tạp, thành tích chập O(k · n · d + n · d bằng với sự kết hợp của lớp tự chú ý và lớp chuyển tiếp theo điểm, cách tiếp cận mà chúng tôi lấy trong mô ^{2}). Tuy nhiên, ngay cả với k = n, độ phức tạp của một hàm có thể tách rời hình của chúng tôi.

Là lợi ích phụ, sự chú ý đến bản thân có thể mang lại nhiều mô hình dễ hiểu hơn. Chúng tôi kiểm tra sự phân phối sự chú ý từ các mô hình của chúng tôi và trình bày cũng như thảo luận về các ví dụ trong phần phụ lục. Không chỉ các đầu chú ý của cá nhân học rõ ràng để thực hiện các nhiệm vụ khác nhau, nhiều người dường như 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 Batching

Chúng tôi đã đào tạo trên bộ dữ liệu tiếng Anh-Đức tiêu chuẩn WMT 2014 bao gồm khoảng 4,5 triệu cặp câu. Các câu được mã hóa bằng cách sử dụng mã hóa cặp byte [3], có từ vựng mục tiêu nguồn được chia sẻ khoảng 37000 mã thông báo. Đối với tiếng Anh-Pháp, chúng tôi đã sử dụng bộ dữ liệu Anh-Pháp WMT 2014 lớn hơn đáng kể bao gồm 36 triệu câu và chia các mã thông báo thành một từ vựng gồm 32000 từ [38]. Các cặp câu được gộp lại với nhau theo độ dài trình tự gần đúng. Mỗi lô đào tạo chứa một tập hợp các cặp câu chứa khoảng 25000 mã thông báo nguồn và 25000 mã thông báo đích.

5.2 Phần cứng và Lịch biểu

Chúng tôi đã đào tạo các mô hình của mình trên một máy có 8 GPU NVIDIA P100. Đối với các mô hình cơ sở của chúng tôi sử dụng siêu tham số được mô tả trong toàn bộ bài báo, mỗi bước đào tạo mất khoảng 0,4 giây. Chúng tôi đã đào tạo các mô hình cơ sở với tổng số 100.000 bước hoặc 12 giờ. Đối với các mô hình lớn của chúng tôi, (được mô tả ở dòng dưới cùng của bảng 3), thời gian bước là 1,0 giây. Các mô hình lớn đã được đào tạo trong 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 = 10 $^{-9}$. Chúng tôi đã thay đổi tỷ lệ học tập trong quá trình đào tạo, theo công thức:

$$\text{lrate} = \text{mô}_{\text{hình}}^{-0,5} \cdot \text{tối thiểu}(\text{step\_num}^{-0.5}, \text{step\_num} \cdot \text{warmup\_steps}^{-1.5}) \quad (3)$$

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

5.4 Chính quy hóa

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

Bỏ học phần còn lại Chúng tôi áp dụng dropout [33] cho đầu ra của mỗi lớp phụ, trước khi nó được thêm vào đầu vào của lớp phụ và được chuẩn hóa. Ngoài ra, chúng tôi áp dụng loại bỏ đối với tổng của các lần nhúng và mã hóa vị trí trong cả ngăn xếp bộ mã hóa và bộ giải mã. Đối với mô hình cơ sở, chúng tôi sử dụng tỷ lệ Pdrop = 0,1.

Bảng 2: Transformer đạt đư ợc điểm số BLEU cao hơn so với các kiểu máy hiện đại nhất trư ớc đây trong các bài kiểm tra từ Anh sang Đức và Anh sang Pháp newstest2014 với chi phí đào tạo chỉ bằng một phần nhỏ.

Ngư ời mẫu	MÀU XANH		Chi phí đào tạo (FLOP)	
	EN-DE	EN-FR	23,75 EN-DE	EN-FR
ByteNet [18]		39.2		1.0 · 1020
Deep-Att + PosUnk [39]	24,6	39,92	2.3 · 1019	1.4 · 1020 9.6
GNMT+RL [38]	25,16	40,46	· 1018	1.5 · 1020 2.0 ·
Chuyển đổi2S [9]	26,03	40,56	1019	1.2 · 1020 8.0 ·
MoE [32]		40,4		1020 1.8 ·
Deep-Att + PosUnk Ensemble [39]	26.30	41,16	1020	1.1 · 1021 7.7 ·
Tổ hợp GNMT + RL [38]	26.36	41,29	1019	1.2 · 1021 3 .3 ·
Nhóm ConvS2S [9]	27,3	38,1		1018 2.3 ·
Máy biến áp (kiểu cơ bản)	28,4	41,8		1019
Máy biến áp (lớn)				

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

6 kết quả

6.1 Dịch máy

Trong nhiệm vụ dịch từ tiếng Anh sang tiếng Đức của WMT 2014, mô hình máy biến áp lớn (Máy biến áp (lớn) trong Bảng 2) hoạt động tốt hơn các mô hình đư ợc báo cáo trư ớc đó (bao gồm cả cụm) hơn 2.0 BLEU, thiết lập một trạng thái mới nhất. điểm BLEU nghệ thuật là 28,4. Cấu hình của mô hình này đư ợc liệt kê ở dòng dư ới cùng của Bảng 3. Quá trình đào tạo mất 3,5 ngày trên 8 GPU P100. Ngay cả mô hình cơ sở của chúng tôi cũng vư ợt qua tất cả các mô hình và tổ hợp đã xuất bản trư ớc đó, với chi phí đào tạo chỉ bằng một phần nhỏ so với bất kỳ mô hình cạnh tranh nào.

Trong nhiệm vụ dịch từ tiếng Anh sang tiếng Pháp của WMT 2014, mô hình lớn của chúng tôi đạt đư ợc số điểm BLEU là 41,0, vư ợt trội so với tất cả các mô hình đơn lẻ đã xuất bản trư ớc đó, với chi phí đào tạo thấp hơn 1/4 so với mô hình hiện đại trư ớc đó. ngư ời mẫu. Mô hình Transformer (lớn) đư ợc đào tạo từ tiếng Anh sang tiếng Pháp đã 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 đư ợc bằng cách lấy trung bình 5 điểm kiểm tra gần nhất, đư ợc viết cách nhau 10 phút. Đối với các mô hình lớn, chúng tôi lấy trung bình 20 điểm kiểm tra cuối cùng. Chúng tôi đã sử dụng tìm kiếm chùm tia với kích thư ớc chùm tia là 4 và hình phạt chiều dài α = 0,6 [38]. Các siêu đư ờng kính này đã đư ợc chọn sau khi thử nghiệm trên bộ phát triển. Chúng tôi đặt độ dài đầu ra tối đa trong quá trình suy luận thành độ dài đầu vào + 50, như ng kết thúc sớm khi có thể [38].

Bảng 2 tóm tắt các kết quả của chúng tôi và so sánh chất lư ợng dịch thuật cũng như chi phí đào tạo của chúng tôi với các kiến trúc mô hình khác từ tài liệu. Chúng tôi ư ớc tính số lư ợng hoạt động dấu phẩy động đư ợc sử dụng để đào tạo một mô hình bằng cách nhân thời gian đào tạo, số lư ợng GPU đư ợc sử dụng và ư ớc tính khả năng duy trì dấu phẩy động có độ chính xác đơn của mỗi GPU [5].

6.2 Biến thể mô hình

Để đánh giá tầm quan trọng của các thành phần khác nhau của Transformer, chúng tôi đã thay đổi mô hình cơ sở của mình theo những cách khác nhau, đo lư ờng sự thay đổi về hiệu suất đối với bản dịch tiếng Anh sang tiếng Đức trên bộ phát triển, newstest2013. Chúng tôi đã sử dụng tìm kiếm theo chùm như đư ợc mô tả trong phần trư ớc, như ng không lấy trung bình điểm kiểm tra. Chúng tôi trình bày những kết quả này trong Bảng 3.

Trong các hàng của Bảng 3 (A), chúng tôi thay đổi số lư ợng đầu chú ý cũng như các thứ nguyên giá trị và khóa chú ý, giữ cho lư ợng tính toán không đổi, như đư ợc mô tả trong Phần 3.2.2. Mặc dù sự chú ý của một đầu là 0,9 BLEU kém hơn so với cài đặt tốt nhất, như ng chất lư ợng cũng giảm xuống với quá nhiều đầu.

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

Bảng 3: Các biến thể về kiến trúc Máy biến áp. Các giá trị không công khai giống hệt với các giá trị của mô hình cơ sở. Tất cả các số liệu đều có trong bộ phát triển dịch từ tiếng Anh sang tiếng Đức, newstest2013. Các vấn đề phức tạp được liệt kê là theo từng từ, theo mã hóa cặp byte của chúng tôi và không được so sánh với các vấn đề phức tạp trên mỗi từ.

	N mô hình	dff	h	dk	dv	Pdrop	ls	đào tạo	tham số	PPL (dev)	BLEU (dev)	×106 bước
cơ sở	6	2048	8	64	64	0,1	0,1	100K	4,92	25,8	24,9	5,29
			1	512	512	4	128			5,00		
			128	16	32	32	32			4,91		
(MỘT)			16	16						5,01		
(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	128			5.75	24.5	28
		1024			128					4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)						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 theo vị trí thay vì hình sin		lớn	6	1024	4096	16			4.92	25,7	
						0,3		300K	4.33		26,4	213

Bảng 4: Transformer tổng quát hóa tốt việc phân tích cú pháp khu vực bầu cử bằng tiếng Anh (Kết quả ở Mục 23 của WSJ)

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

Trong Bảng 3 hàng (B), chúng tôi quan sát thấy rằng việc giảm kích thước khóa chú ý dk làm ảnh hưởng đến chất lượng mô hình. Điều này cho thấy rằng việc xác định khả năng tương thích là không dễ dàng và chức năng tương thích tinh vi hơn sản phẩm chấm có thể có lợi. Chúng tôi quan sát thêm ở các hàng (C) và (D) rằng, như mong đợi, các mô hình lớn hơn sẽ tốt hơn và việc bỏ học rất hữu ích trong việc tránh bị quá khớp. Trong hàng (E), chúng tôi thay thế mã hóa vị trí hình sin bằng các nhúng vị trí đã học [9] và quan sát các kết quả gần như giống hệt với mô hình cơ sở.

6.3 Phân tích cú pháp bầu cử tiếng Anh

Để đánh giá xem Transformer có thể khái quát hóa thành các tác vụ khác hay không, chúng tôi đã thực hiện các thử nghiệm về phân tích cú pháp khu vực bầu cử bằng tiếng Anh. Nhiệm vụ này đặt ra những thách thức cụ thể: đầu ra phải chịu sự tác động mạnh mẽ của cấu trúc

ràng buộc và dài hơn đáng kể so với đầu vào. Hơn nữa, các mô hình tuần tự theo trình tự RNN đã không thể đạt được kết quả tiên tiến nhất trong các chế độ dữ liệu nhỏ [37].

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

Chúng tôi chỉ thực hiện một số thử nghiệm nhỏ để chọn người bỏ học, cả mức độ chú ý và phần 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 không thay đổi so với mô hình dịch cơ sở từ tiếng Anh sang tiếng Đức. Trong quá trình suy luận, chúng tôi đã tăng độ dài đầu ra tối đa thành độ dài đầu vào + 300. Chúng tôi đã sử dụng kích thước chùm tia là 21 và α = 0,3 chỉ cho cả WSJ và cài đặt bán giám sát.

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

Trái ngược với các mô hình tuần tự theo trình tự RNN [37], Transformer hoạt động tốt hơn Trình phân tích cú pháp Berkeley [29] ngay cả khi chỉ đào tạo trên tập huấn luyện WSJ gồm 40 nghìn câu.

7. Kết luận

Trong công việc này, chúng tôi đã trình bày Transformer, mô hình tải nạp trình tự đầu tiên hoàn toàn dựa trên sự chú ý, thay thế các lớp lặp lại được sử dụng phổ biến nhất trong kiến trúc bộ mã hóa-giải mã bằng khả năng tự chú ý nhiều đầu.

Đối với các tác vụ dịch thuật, Transformer có thể được đào tạo nhanh hơn đáng kể so với các kiến trúc dựa trên các lớp tích chập hoặc hồi quy. Trên cả hai tác vụ dịch từ tiếng Anh sang tiếng Đức và WMT 2014 từ tiếng Anh sang tiếng Pháp, chúng tôi đạt được một trình độ nghệ thuật mới. Trong nhiệm vụ trước đây, mô hình tốt nhất của chúng tôi vượt trội hơn cả tất cả các nhóm được báo cáo trước đó.

Chúng tôi rất vui mừng về tương lai của các mô hình dựa trên sự chú ý và dự định áp dụng chúng cho các nhiệm vụ khác. Chúng tôi dự định mở rộng Máy biến áp cho các vấn đề liên quan đến phương thức đầu vào và đầu ra ngoài văn bản và điều tra các cơ chế tập trung hạn chế, cục bộ để xử lý hiệu quả các đầu vào và đầu ra lớn như hình ảnh, âm thanh và video. Làm cho việc tạo ít trình tự hơn là một mục tiêu nghiên cứu khác của chúng tôi.

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

Lời cảm ơn Chúng tôi rất biết ơn Nal Kalchbrenner và Stephan Gouws vì những nhận xét, chỉnh sửa và nguồn cảm hứng hiệu quả của họ.

Người giới thiệu

[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 bằng cách cùng nhau học cách căn chỉnh và dịch. CoRR, abs/1409.0473, 2014.

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

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

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

[6] Francois Chollet. Ngoại lệ: Học sâu với các cấu trúc có thể tách rời theo chiều sâu. arXiv bản in trước arXiv:1610.02357, 2016.

[7] Junyoung Chung, Ç aglar Gülçehre, Kyunghyun Cho, và Yoshua Bengio. Đánh giá thực nghiệm các mạng thần kinh tái phát có kiểm soát trên mô hình trình tự. CoRR, abs/1412.3555, 2014.

[8] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, và Noah A. Smith. Văn phạm mạng thần kinh tái phát . Ở Proc. của NAACL, 2016.

[9] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, và Yann N. Dauphin. Trình tự phức hợp để học trình tự. bản in trư ớc arXiv arXiv:1705.03122v2, 2017.

[10] Alex Graves. Tạo trình tự với các mạng thần kinh tái phát. bản in trư ớc arXiv arXiv:1308.0850, 2013.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, và Jian Sun. Học sâu còn lại để nhận dạng tuổi im. Trong Kỷ yếu của Hội nghị IEEE về Tầm nhìn Máy tính và Nhận dạng Mẫu, trang 770–778, 2016.

[12] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, và Jürgen Schmidhuber. Dòng chuyển màu trong mạng lặp lại: khó khăn trong việc học các 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 đào tạo với các chú thích tiềm ẩn trên các ngôn ngữ. Trong Kỷ yếu của Hội nghị năm 2009 về các Phư ơng 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 các giới hạn của mô hình hóa ngôn ngữ. bản in trư ớc arXiv arXiv:1602.02410, 2016.

[16] Łukasz Kaiser và Samy Bengio. Bộ nhớ tích cực có thể thay thế sự chú ý? 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. in sẵn arXiv arXiv:1610.10099v2, 2017.

[19] Yoon Kim, Carl Denton, Lư ơng Hoàng, và Alexander M. Rush. Mạng lư ới chú ý có cấu trúc. Trong Hội nghị Quốc tế về Đại diện Học tập, 2017.

[20] Diederik Kingma và Jimmy Ba. Adam: Một phư ơng pháp tối ư u hóa ngẫu nhiên. Trong ICLR, 2015.

[21] Oleksii Kuchaiev và Boris Ginsburg. Thủ thuật nhân tố hóa cho các mạng LSTM. bản in trư ớc arXiv arXiv:1703.10722, 2017.

[22] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou và Yoshua Bengio. Một cấu trúc nhúng câu tự chú ý. bản in trư ớc arXiv arXiv:1703.03130, 2017.

[23] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Trình tự đa tác vụ để học theo trình tự. bản in trư ớc arXiv arXiv:1511.06114, 2015.

[24] Lư ơng Minh-Thăng, Hiếu Phạm, và Christopher D Manning. Các phư ơng pháp hiệu quả để dịch máy thần kinh dựa trên sự chú ý. bản in trư ớc arXiv arXiv:1508.04025, 2015.

[25] Mitchell P Marcus, Mary Ann Marcinkiewicz, và Beatrice Santorini. Xây dựng kho ngữ liệu tiếng Anh có chú thích lớn : The penn treebank. 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 Kỷ yếu Hội nghị Công nghệ Ngôn ngữ Con ngư ời 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 hủy. Trong Phư ơng pháp thực nghiệm trong xử lý ngôn ngữ tự nhiên, 2016.

[28] Romain Paulus, Caiming Xiong, và Richard Socher. Một mô hình củng cố sâu cho trừu tư ợng tóm tắt. bản in trư ớc arXiv arXiv:1705.04304, 2017.

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

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

[31] Rico Sennrich, Barry Haddow, và Alexandra Birch. Dịch máy thần kinh của các từ hiếm với các đơn vị từ phụ. bản in trư ớc arXiv arXiv:1508.07909, 2015.

[32] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, và Jeff Dean. Mạng lư ới thần kinh cực kỳ lớn: Lớp hỗn hợp các chuyên gia đư ợc kiểm soát thư a thớt . bản in trư ớc arXiv arXiv:1701.06538, 2017.

[33] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, và Ruslan Salakhutdi tiểu thuyết. Bỏ học: một cách đơn giản để ngăn mạng nơ-ron khớp quá mức. Tạp chí Nghiên cứu Máy học, 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. Hiệp hội Curran, Inc., 2015.

[35] Ilya Sutskever, Oriol Vinyals, và Quoc VV Le. Trình tự để học chuỗi với mạng lư ới thần kinh. Trong Những tiến bộ trong Hệ thống xử lý thông tin thần kinh, trang 3104–3112, 2014.

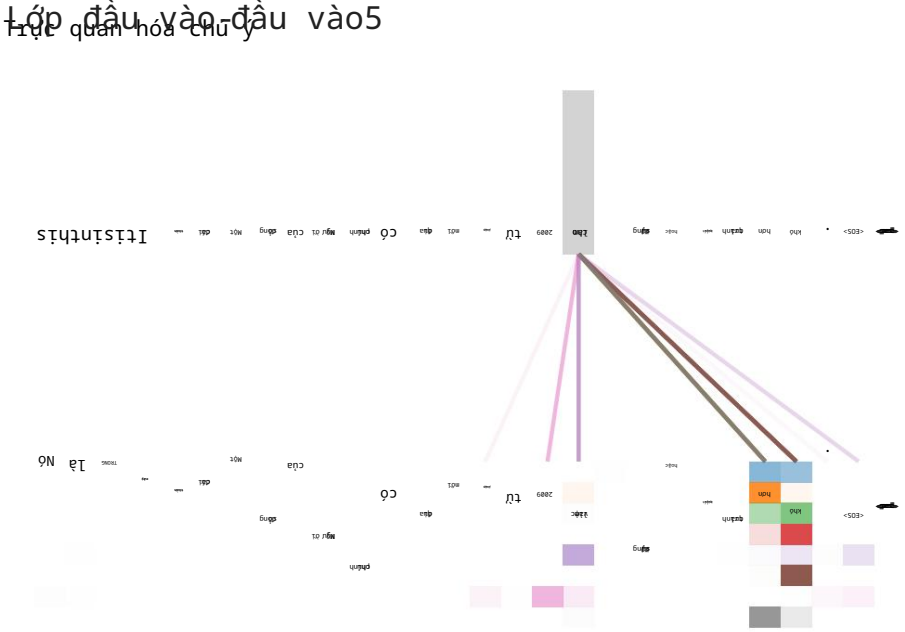
[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, và Zbigniew Wojna. Suy nghĩ lại về kiến trúc ban đầu cho thị giác máy tính. CoRR, abs/1512.00567, 2015.

[37] Vinyals & Kaiser, Koo, Petrov, Sutskever, và Hinton. Ngữ pháp như một ngoại ngữ. 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 ngư ời và máy. bản in trư ớc arXiv arXiv:1609.08144, 2016.

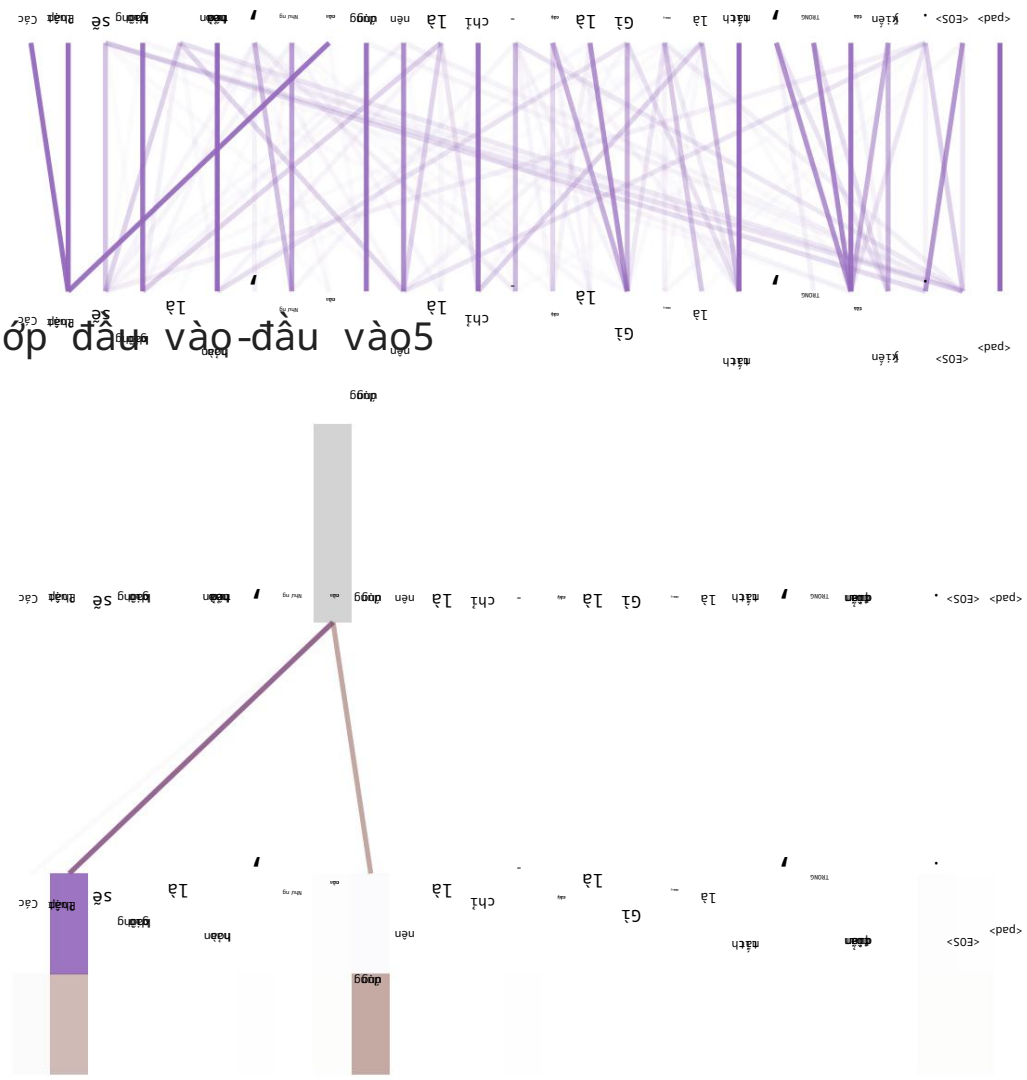
[39] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, và Wei Xu. Các mô hình lặp lại sâu với các kết nối chuyển tiếp nhanh để dịch máy thần kinh. CoRR, abs/1606.04199, 2016.

[40] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, và Jingbo Zhu. Phân tích cú pháp thành phần shift-reduce nhanh chóng và chính xác . Trong Kỷ yếu Hội nghị thư ờng niên lần thứ 51 của ACL (Tập 1: Các bài viết dài), trang 434–443. ACL, tháng 8 năm 2013.



Hình 3: Một ví dụ về cơ chế chú ý tuân theo các phụ thuộc khoảng cách xa trong khả năng tự chú ý của bộ mã hóa ở lớp 5/6. Nhiều phần chú ý chú ý đến một phụ thuộc xa của động từ 'tạo', hoàn thành cụm từ 'tạo.. .khó hơn'. Chú ý ở đây chỉ hiển thị cho từ 'làm'. Màu sắc khác nhau đại diện cho đầu khác nhau. Xem tốt nhất trong màu sắc.

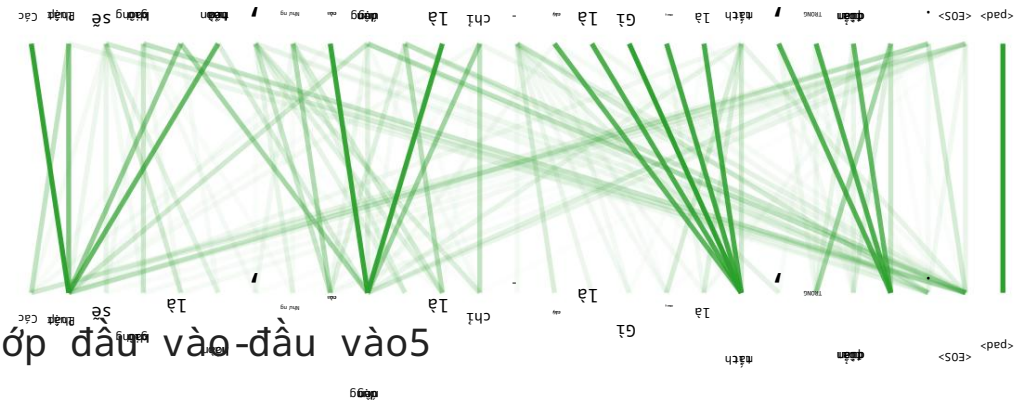
Lớp đầu vào-đầu vào5

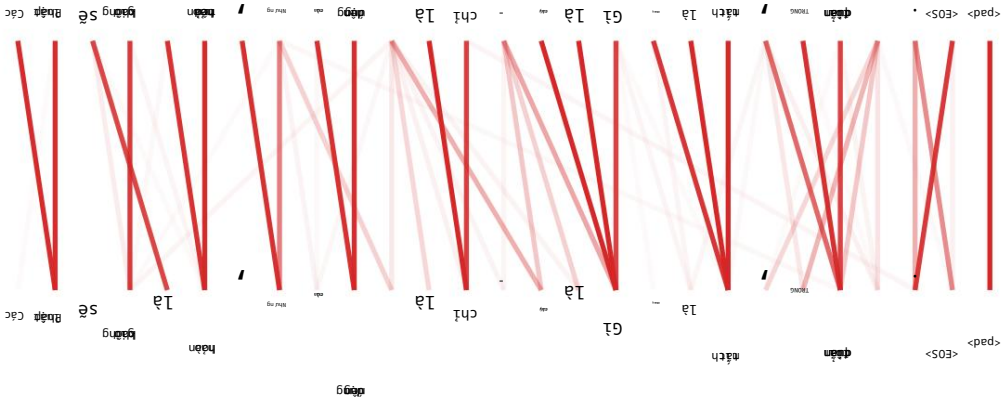


Lớp đầu vào-đầu vào5

Hình 4: Hai đầu chú ý, cũng ở lớp 5/6, dư ờng như liên quan đến việc giải quyết anaphora. Trên cùng: Chú ý đầy đủ cho đầu 5. Dư ới cùng: Chú ý riêng biệt chỉ từ 'its' cho đầu chú ý 5 và 6. Lư u ý rằng các chú ý rất sắc nét đối với từ này.

Lớp đầu vào-đầu vào5





Hình 5: Nhiều ngư ời trong số những ngư ời đứng đầu sự chú ý thể hiện hành vi có vẻ liên quan đến cấu trúc của câu. Chúng tôi đư a ra hai ví dụ như vậy ở trên, từ hai đầu khác nhau từ bộ mã hóa tự chú ý ở lớp 5/6. Các đầu rõ ràng đã học để thực hiện các nhiệm vụ khác nhau.