TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

-----□□□□□-----

KHÓA LUẬN TỐT NGHIỆP

NGÀNH KỸ THUẬT DỮ LIỆU

Đề Tài:

SỬ DỤNG MÔ HÌNH MẠNG HỒI QUY TRUY HỒI ĐỂ TẠO CHÚ THÍCH CHO HÌNH ẢNH

GIẢNG VIÊN HƯỚNG DẪN

ThS. Quách Đình Hoàng

SVTH: Đinh Quốc Hùng – 19133025

Trần Nguyên Thái Bảo – 19133010

KHÓA 2019 - 2023

ĐẠI HỌC SƯ PHẠM KỸ THUẬT
TP. HCM
KHOA CNTT
****

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh Phúc
****

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên 1: **Đinh Quốc Hùng** MSSV 1: **19133025**

Họ và tên Sinh viên 2: **Trần Nguyên Thái Bảo** MSSV 2: **19133010**

Ngành: **Kỹ thuật dữ liệu**

Tên đề tài: **SỬ DỤNG MÔ HÌNH MẠNG HỒI QUY TRUY HỒI ĐỂ TẠO CHÚ THÍCH CHO HÌNH ẢNH**

Họ và tên Giáo viên hướng dẫn: **ThS. Quách Đình Hoàng**

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đánh giá loại:

5. Điểm:

Tp. Hồ Chí Minh, ngày tháng năm 2021

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên nhóm xin phép được gửi lời cảm ơn chân thành và sâu sắc nhất đến với Khoa Công Nghệ Thông Tin – Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh đã tạo điều kiện cho nhóm chúng em được học tập, phát triển nền tảng kiến thức sâu sắc và thực hiện đề tài này.

Bên cạnh đó nhóm chúng em xin gửi đến thầy Quách Đình Hoàng lời cảm ơn sâu sắc nhất. Trải qua một quá trình dài học tập và thực hiện đề tài trong thời gian qua. Thầy đã tận tâm chỉ bảo nhiệt tình nhóm chúng em trong suốt quá trình từ lúc bắt đầu cũng như kết thúc đề tài này.

Với sự hướng dẫn nhiệt tình, giảng dạy tận tình đầy đủ kiến thức của thầy Quách Đình Hoàng, chúng em đã học tập và hiểu được những kiến thức về các thuật toán trong bài toán sử dụng mô hình hồi quy để tạo chú thích cho hình ảnh

Tuy nhiên lượng kiến thức là vô tận và với khả năng hạn hẹp chúng em đã rất cố gắng để hoàn thành một cách tốt nhất. Chính vì vậy việc xảy ra những thiếu sót là điều khó có thể tránh khỏi. Chúng em hi vọng nhận được sự góp ý tận tình của quý thầy (cô) qua đó chúng em có thể rút ra được bài học kinh nghiệm và hoàn thiện và cải thiện nâng cấp lại sản phẩm của mình một cách tốt nhất có thể.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

LỜI CẢM ƠN 3

PHẦN MỞ ĐẦU 6

1. Tính cấp thiết của đề tài 6

2. Đối tượng nghiên cứu 6

3. Phạm vi nghiên cứu 6

4. Kết quả dự kiến đạt được 6

PHẦN NỘI DUNG 7

CHƯƠNG 1: TỔNG QUAN VỀ HỌC SÂU 7

1.1. Tổng quan về học sâu 7

1.2. Mạng neural nhân tạo 8

1.2.1. Mạng neural đơn giản 8

1.2.2. Kiến trúc của mạng neural và mạng neural lan truyền thuận đa tầng 8

1.2.2.1 Kiến trúc của mạng neural 8

1.2.2.2. Mạng neural lan truyền thuận đa tầng 9

1.2.3. Tầng ẩn(hidden layer) và hàm kích hoạt(Activation function) 9

1.2.3.1. ReLU 10

1.2.3.2. Sigmoid và Tanh 11

1.2.4. Thiết kế kiến trúc mạng 12

1.3. Tổng quan về thị giác máy tính 13

1.3.1. Giới thiệu về thị giác máy tính 13

1.3.2. Các thành phần cơ bản trong thị giác máy tính 14

1.3.3. Ứng dụng của thị giác máy tính 14

CHƯƠNG 2: MẠNG HỒI QUY TRUY HỒI - RECURRENT NEURAL NETWORK VÀ MẠNG TÍCH CHẬP - CONVOLUTION NEURAL NETWORK 15

2.1. Mạng hồi quy truy hồi (Recurrent Neural Network) 15

2.2. Mô hình bài toán mạng thần kinh hồi quy (mô hình RNN, loss function) 15

2.2.1. Dữ liệu dạng chuỗi 15

2.2.2. Hàm mất mát(Loss function) và hàm kích hoạt(Activation function) 17

2.2.3. Lan truyền ngược(Back-Propagation) 20

2.2.4. Long Short-Term Memory (Long Short-Term Memory) 21

2.4. Mạng tích chập (tích chập) 22

2.4.1. Inception Model 23

2.4.2. Inception Version 2 26

2.4.3. Inception V3 28

2.4.4. Inception-V4 and Inception-ResNets 29

CHƯƠNG 3: TỔNG QUAN VỀ BÀI TOÁN TẠO CHÚ THÍCH HÌNH ẢNH (IMAGE CAPTIONING) 32

3.1. Tổng quan về bài toán tạo chú thích cho hình ảnh 32

3.1.1. Khái niệm 32

3.1.2. Lịch sử phát triển của bài toán tạo chú thích hình ảnh 32

3.2. Các phương pháp tiếp truyền thống và các mô hình deep learning được áp dụng cho bài toán 32

3.3. Các thách thức của bài toán 33

3.4. Tiền xử lý dữ liệu 34

3.5. Sử dụng mạng tích chập để trích xuất đặc trưng 34

3.6. Sử dụng mạng thần kinh hồi quy để sinh ra câu chú thích 34

3.7. Huấn luyện mô hình 34

3.8. Đánh giá mô hình 34

3.8. Hạn chế của mô hình mạng thần kinh hồi quy 36

3.9. Hạn chế của Long Short-Term Memory 38

CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT QUẢ ĐẠT ĐƯỢC 39

4.1.Tập dữ liệu 39

4.2. Tiền xử lý dữ liệu 39

4.3. Ứng dụng mạng tích chập để trích xuất đặc trưng 39

4.4. Ứng dụng mạng thần kinh hồi quy để tạo chú thích hình ảnh 39

4.5. Huấn luyện mô hình **40**

4.6. Đánh giá mô hình **40**

4.7. Kết quả 40

PHẦN KẾT LUẬN **42**

1. Tổng kết **42**

2. Hạn chế của đề tài **42**

3. Hướng phát triển **42**

TÀI LIỆU THAM KHẢO **43**

PHẦN MỞ ĐẦU

1. Tính cấp thiết của đề tài

Bài toán image captioning là một trong những bài toán quan trọng của lĩnh vực thị giác máy tính và xử lý ngôn ngữ tự nhiên. Việc nghiên cứu và phát triển các giải pháp cho bài toán này sẽ đóng góp quan trọng vào các ứng dụng thực tế như hỗ trợ mô tả hình ảnh, tự động viết phụ đề cho video, hỗ trợ người dùng khi tìm kiếm thông tin, hỗ trợ đọc cho người khiếm thị,.... Từ đó, đề tài tìm hiểu về bài toán image captioning được đánh giá là rất cấp thiết và có tính ứng dụng cao trong thực tế.

2. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài tìm hiểu về bài toán image captioning là các phương pháp, mô hình và công cụ được sử dụng để giải quyết bài toán này. Đối tượng này bao gồm các mô hình Deep Learning như Convolutional Neural Networks (tích chập) và Recurrent Neural Networks (mạng thần kinh hồi quy), các thuật toán xử lý ngôn ngữ tự nhiên và các công cụ như thư viện mã nguồn mở Tensorflow, PyTorch và Keras. Ngoài ra, đối tượng nghiên cứu còn bao gồm các tập dữ liệu được sử dụng để huấn luyện và đánh giá mô hình, tổng hợp mô tả hình ảnh, ...

3. Phạm vi nghiên cứu

Tìm hiểu về các phương pháp trích xuất đặc trưng và mô hình học sâu được sử dụng để giải quyết bài toán image captioning. Nghiên cứu về các tập dữ liệu được sử dụng để huấn luyện và đánh giá hiệu quả của các mô hình image captioning. Xây dựng và huấn luyện một mô hình image captioning sử dụng một trong các phương pháp trích xuất đặc trưng và mô hình học sâu được tìm hiểu. Đánh giá hiệu quả của mô hình đã được xây dựng

4. Kết quả dự kiến đạt được

Kết quả dự kiến đạt được của đề tài tìm hiểu về bài toán image captioning là hiểu rõ về bài toán này, các phương pháp và kỹ thuật hiện đang được sử dụng để giải quyết vấn đề này. Đề tài cũng sẽ thực hiện cài đặt và đánh giá hiệu quả của phương pháp image captioning trên các tập dữ liệu ảnh. Kết quả của khóa luận sẽ giúp cho các nhà nghiên cứu trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo có thêm kiến thức và kinh nghiệm trong giải quyết bài toán image captioning.

CHƯƠNG 1: TỔNG QUAN VỀ HỌC SÂU

1.1. Tổng quan về học sâu

Học sâu là một nhánh của machine learning và trí tuệ nhân tạo và ngày nay học sâu được xem như là một công nghệ cốt lõi của thời đại công nghiệp 4.0. Với khả năng học từ các dữ liệu và đưa ra độ chính xác cao, học sâu được ứng dụng rộng rãi trong các lĩnh vực khác nhau của đời sống như sức khỏe, nhận dạng hình ảnh, phân tích văn bản, an ninh và ngày nay càng được ứng dụng rộng rãi hơn thế nữa. Tuy nhiên, việc xây dựng các mô hình học sâu phù hợp với các nhiệm vụ mang tính thách thức thì rất khó. Việc thiếu hiểu biết về các kiến thức cốt lõi dẫn tới việc biến học sâu thành một cái hộp đen ở đó ta chỉ sử dụng nó như một công cụ nhưng không hiểu rõ bản chất của nó là gì, điều này có thể cản trở ta phát triển nó.

Học sâu có thể nói là một tập con của học máy, học sâu dựa trên mạng thần kinh nhân tạo với nhiều lớp khác nhau còn được gọi là mạng thần kinh sâu(Deep neural network). Các mạng thần kinh này được lấy cảm hứng từ cấu trúc và chức năng của bộ não con người và chúng được thiết kế để học với một lượng dữ liệu lớn theo cách có giám sát và không có giám sát.

Trong thập kỷ vừa qua, học sâu dần trở thành một xu hướng với sự quay lại mạnh mẽ của mình, học sâu ngày càng thu hút sự quan tâm và trở thành một mảng lớn trong trí tuệ nhân tạo ngày nay.

1.2. Mạng neural nhân tạo

1.2.1. Mạng neural đơn giản

Kiến trúc của mạng thần kinh được tạo thành từ lớp đầu vào, đầu ra và lớp ẩn. Bản thân mạng nơ-ron, hay mạng nơ-ron nhân tạo (ANN), là một tập hợp con của máy học được thiết kế để bắt chước sức mạnh xử lý của bộ não con người. Mạng nơ-ron hoạt động bằng cách truyền dữ liệu qua các lớp của nơ-ron nhân tạo.

Có nhiều thành phần cho một kiến trúc mạng thần kinh. Mỗi mạng thần kinh có một vài thành phần chung:

Đầu vào(Input) - Đầu vào là dữ liệu được đưa vào mô hình để phục vụ cho mục đích học tập và đào tạo. Ví dụ: đầu vào trong phát hiện đối tượng có thể là một mảng các giá trị pixel liên quan đến một hình ảnh.

Trọng số(Weight) - Trọng số giúp tổ chức các biến theo tầm quan trọng và tác động của đóng góp.

Hàm truyền(Transfer function) - Công việc của hàm truyền là kết hợp nhiều đầu vào thành một giá trị đầu ra để có thể áp dụng hàm kích hoạt. Nó được thực hiện bằng một tổng đơn giản của tất cả các đầu vào cho hàm truyền.

Hàm kích hoạt(Activation function) - Vai trò của chức năng kích hoạt là quyết định có nên kích hoạt một neural cụ thể hay không. Quyết định này dựa trên việc liệu đầu vào của neural có quan trọng đối với quá trình dự đoán hay không.

Bias - Vai trò của bias là thay đổi giá trị do hàm kích hoạt tạo ra. Vai trò của nó tương tự như vai trò của hằng số trong hàm tuyến tính.

1.2.2. Kiến trúc của mạng neural và mạng neural lan truyền thuận đa tầng

1.2.2.1 Kiến trúc của mạng neural

Lớp đầu vào(Input Layer) - Dữ liệu mà chúng ta cung cấp cho mô hình được tải vào lớp đầu vào từ các nguồn bên ngoài như tệp CSV hoặc dịch vụ web. Đây là lớp hiển thị duy nhất trong kiến trúc Mạng nơ-ron hoàn chỉnh truyền toàn bộ thông tin từ thế giới bên ngoài mà không cần bất kỳ tính toán nào.

Lớp ẩn(Hidden layer) - Các lớp ẩn là thứ tạo nên deep learning như ngày nay. Chúng là các lớp trung gian thực hiện tất cả các tính toán và trích xuất các tính năng từ dữ liệu. Có thể có nhiều lớp ẩn được kết nối với nhau để tìm kiếm các tính năng ẩn khác nhau trong dữ liệu. Ví dụ: trong xử lý ảnh, các lớp ẩn đầu tiên chịu trách nhiệm về các tính năng cấp cao hơn như các cạnh, hình dạng hoặc ranh giới. Mặt khác, các lớp ẩn sau thực hiện các tác vụ phức tạp hơn như xác định các đối tượng hoàn chỉnh (ô tô, tòa nhà, con người).

Lớp đầu ra(Output layer) - Lớp đầu ra lấy đầu vào từ các lớp ẩn trước đó và đưa ra dự đoán cuối cùng dựa trên kết quả học tập của mô hình. Đây là lớp quan trọng nhất nơi chúng ta nhận được kết quả cuối cùng. Trong trường hợp mô hình phân loại/hồi quy, lớp đầu ra thường có một nút duy nhất. Tuy nhiên, nó hoàn toàn dành riêng cho vấn đề và phụ thuộc vào cách xây dựng mô hình.

1.2.2.2. Mạng neural lan truyền thuận đa tầng

Mạng Lan Truyền Thuận Đa Tầng là một dạng mô hình học sâu điển hình, với mục tiêu là xấp xỉ một hàm f^*, chẳng hạn như một bộ phân loại: $y = f^*(x)$ ánh xạ dữ liệu đầu vào *x* tới một nhãn *y*. Một mạng đa tầng định nghĩa một phép ánh xạ $y = f(x;\theta)$ và học giá trị của các tham số θ để thu được xấp xỉ tốt nhất của ánh xạ thực sự. Những mô hình này được gọi là lan truyền thuận vì chúng nhận thông tin từ đầu vào *x* và lan truyền qua các phép toán trung gian trong mô hình để tới đầu ra y.

1.2.3. Tầng ẩn(hidden layer) và hàm kích hoạt(Activation function)

Chúng ta gọi mạng nơron lan truyền thuận là mạng vì chúng thường được biểu diễn bằng cách hợp thành từ nhiều tầng (layer) khác nhau. Chẳng hạn, ta có 3 hàm số $f^{(1)}$, $f^{(2)}$, $f^{(3)}$ liên kết với nhau thành một chuỗi để tạo thành hàm $f^{(x)} = f^{(3)}(f^{(2)}(f^{(1)}))$. Cấu trúc chuỗi là cấu trúc mạng nơron phổ biến nhất. $f^{(1)}$ được gọi là tầng đầu tiên của mạng nơron, $f^{(2)}$ được gọi là tầng thứ hai, và cứ như vậy. Chiều dài của toàn bộ chuỗi được gọi là độ sâu (depth) của mạng nơron. Tầng cuối cùng của một mạng lan truyền thuận được gọi là tầng đầu ra (output layer).

Dù vậy, nhiều tầng ẩn hơn cũng không làm cho hàm tổng quát trở nên phi tuyến, một điều hạn chế đối với các thuật toán học máy. Nghĩa là chúng vẫn còn quá đơn giản để biểu diễn các bài toán phức tạp (phi tuyến) như xử lý ngôn ngữ tự nhiên. Nhưng có một nguyên tắc chung trong khoa học máy tính là ta có thể xây dựng các hệ thống phức tạp dựa từ các thành phần đơn giản.

Dựa trên cơ sở với nhiều tầng ẩn hidden layer ta có thể kết hợp nhiều hàm tuyến lại với nhau tạo thành một hàm phi tuyến. Các hàm như vậy ta gọi là hàm kích hoạt (activation function).

Bằng cách dùng hàm kích hoạt ta có thể vượt qua những hạn chế của mô hình tuyến tính. Cách dễ nhất để làm điều này là xếp chồng nhiều tầng kết nối đầy đủ lên nhau, trong mỗi tầng ẩn ta có mỗi hàm kích hoạt. Giá trị đầu ra của mỗi tầng được đưa làm giá trị đầu vào cho tầng bên trên, cho đến khi tạo ra kết quả đầu ra.

Các hàm kích hoạt quyết định một nơron có được kích hoạt hay không bằng cách tính tổng có trọng số và cộng thêm hệ số điều chỉnh. Các hàm kích hoạt rất đa

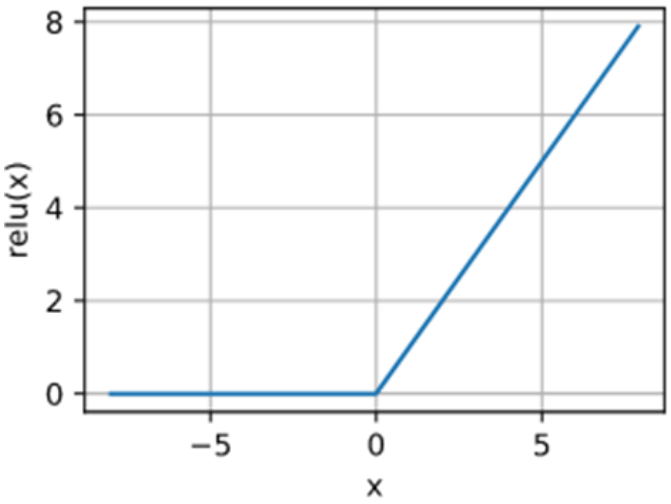
dạng và được phát triển để phù hợp cho từng bài toán cho đến hiện tại có một số hàm kích hoạt phổ biện như ReLU, Sigmoid, Tanh,...

1.2.3.1. ReLU

ReLU sử dụng hàm kích hoạt:

$$g(z) = max\{0, z\} \quad (1.2)$$

ReLU dễ tối ưu vì chúng gần giống đơn vị tuyến tính, ReLU sẽ cho kết quả bằng 0 trên một nửa miền xác định của hàm. Điều này khiến đạo hàm lan truyền qua đơn vị này giữ nguyên độ lớn nếu đơn vị này còn hoạt động.



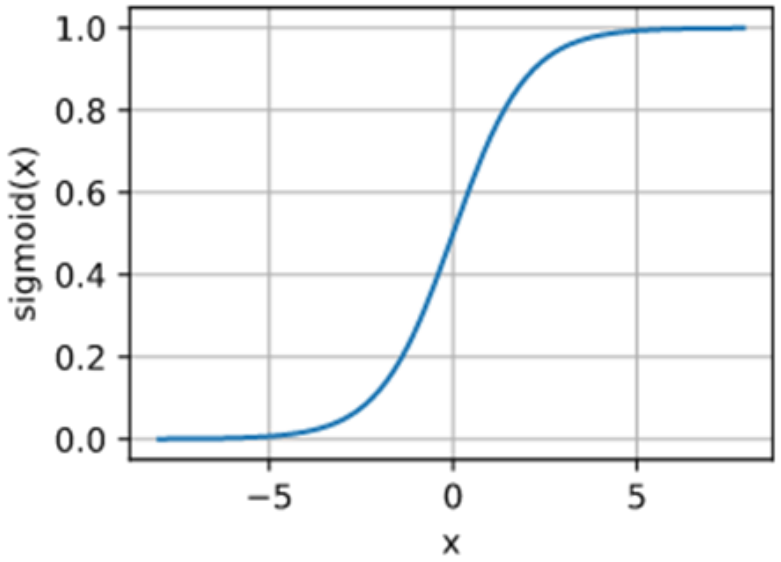
Hình 1. Đồ thị của ReLU [1]

Đây là hàm kích hoạt mặc định được khuyến cáo sử dụng trong hầu hết các mạng nơron lan truyền thuật. ReLU còn có một số dạng khác chủ yếu để khắc phục các nhược điểm của nó như không thể học thông qua các phương pháp dựa trên gradient tại các mẫu huấn luyện có hàm kích hoạt bằng 0.

1.2.3.2. Sigmoid và Tanh

Khi phương pháp học dựa trên gradient trở nên phổ biến, hàm sigmoid là một lựa tốt của đơn vị ngưỡng (threshold) bởi tính liên tục và khả vi tại mọi điểm. Hàm sigmoid là hàm kích hoạt được sử dụng nhiều ở các đơn vị đầu ra, khi ta muốn biểu diễn kết quả đầu ra như là các xác suất của bài toán phân loại nhị phân (tương tự như softmax). Ta có hàm sigmoid như sau:

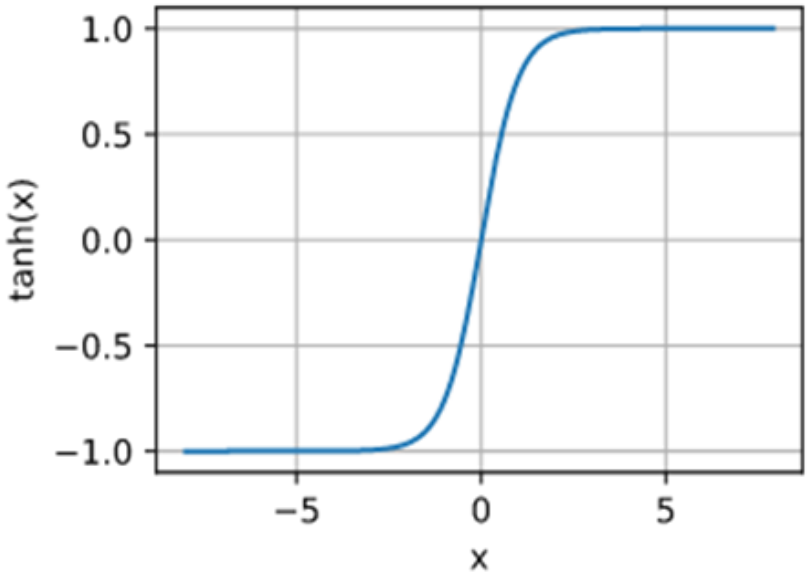
$$g(z) = \sigma(z) = \frac{1}{1+exp(-z)} \quad (1.3)$$



Hình 2. Đồ thị Sigmoid [2]

Trước khi ReLU ra đời, phần lớn các mạng nơron đều sử dụng hàm kích hoạt sigmoid trong các tầng ẩn. Sau này hàm sigmoid hầu hết bị thay thế bằng hàm ReLU vì nó đơn giản hơn và giúp cho việc huấn luyện trở nên dễ dàng hơn. Ta cũng có hàm kích hoạt tanh dựa trên sigmoid:

$$g(x) = tanh(z) = 2\sigma(2z) - 1 \quad (1.4)$$



Hình 3. Đồ thị Tanh [3]

Hàm tanh khá tương đồng với sigmoid nhưng có nó có thẻ mang giá trị âm, ta có thể áp dụng tùy thuộc vào bài toán ta cần dùng

1.2.4. Thiết kế kiến trúc mạng

Một yếu tố then chốt khác trong mạng nơron làm xác định kiến trúc của mạng. Kiến trúc mà ta thiết kế đề cập đến cấu trúc tổng thể của mạng: mạng cần bao nhiêu đơn vị và các đơn vị này sẽ kết nối với nhau như thế nào.

Phần lớn các mạng nơron được tổ chức theo nhóm các đơn vị, được gọi là đa tầng. Đa phần kiến trúc mạng nơron xếp các tầng này thành cấu trúc chuỗi, mỗi tầng là một hàm số của tầng liền trước nó. Trong những kiến trúc dạng chuỗi như vậy, các vấn đề kiến trúc chính cần cân nhắc lựa chọn độ sâu của mạng và chiều rộng của mỗi tầng.

Ta cần một tầng ẩn để khớp với tập huấn luyện, các mạng sâu hơn thường có thể sử dụng số đơn vị trên mỗi tầng và tham số ít hơn rất nhiều, cũng như thường khái quát hóa tốt hơn trên tập kiểm thử, nhưng lại khó tối ưu hơn. Như vậy ta chỉ có thể tìm thấy kiến trúc mạng lý tưởng thông qua thực nghiệm bằng cách quan sát sai số trên tập kiểm định.

1.3. Tổng quan về thị giác máy tính

1.3.1. Giới thiệu về thị giác máy tính

Thị giác máy tính là một lĩnh vực đa ngành cho phép các hệ thống và máy tính rút ra thông tin hữu ích từ các video, hình ảnh kỹ thuật số và các dạng đầu vào trực quan khác. Trí tuệ nhân tạo giúp máy tính suy nghĩ, trong khi thị giác máy tính giúp chúng cảm nhận và hiểu được môi trường xung quanh. Nó mô phỏng mắt con người và được sử dụng để huấn luyện các mô hình thực hiện các chức năng khác nhau với sự trợ giúp của các camera, thuật toán và dữ liệu thay vì các thần kinh quang học, võng mạc và vỏ não thị giác

Thị giác máy tính là một lĩnh vực tương đối trẻ, phát triển trong thế kỷ 20 nhờ vào sự tiến bộ về công nghệ điện tử và máy tính. Các công nghệ và kỹ thuật trong thị giác máy tính phát triển nhanh chóng trong những năm gần đây, đặc biệt là với sự gia tăng về dữ liệu và khả năng tính toán của máy tính. Những bước tiến đáng chú ý trong lịch sử của thị giác máy tính bao gồm:

Những năm 1960: Bắt đầu nghiên cứu về nhận diện khuôn mặt và thị giác máy tính đầu tiên được phát triển.

Những năm 1970: Các kỹ thuật truyền thống như xử lý ảnh số, xử lý dấu vết và nhận dạng mẫu được áp dụng trong thị giác máy tính.

Những năm 1980: Những kỹ thuật học máy đầu tiên được áp dụng trong thị giác máy tính. Các phương pháp học máy như máy vector hỗ trợ (SVM) và mạng nơ-ron nhân tạo (ANN) được phát triển.

Những năm 1990: Việc ứng dụng thị giác máy tính trong các lĩnh vực như y tế, an ninh và công nghiệp bắt đầu gia tăng.

Những năm 2000: Sự phát triển của Internet và tăng cường về khả năng tính toán đã đẩy mạnh sự phát triển của thị giác máy tính. Các phương pháp học sâu, đặc biệt là mạng nơ-ron tích chập (tích chập), đã được phát triển và áp dụng rộng rãi.

Những năm 2010: Thị giác máy tính đã đạt được nhiều thành tựu đáng kể, bao gồm khả năng nhận diện khuôn mặt, phát hiện vật thể và phân loại hình ảnh với độ chính xác cao hơn cả con người. Công nghệ thị giác máy tính được sử dụng rộng rãi trong các lĩnh vực như tự động hóa công nghiệp, xe tự lái và thị trường quảng cáo.

1.3.2. Các thành phần cơ bản trong thị giác máy tính

Xử lý ảnh: Thao tác với ảnh số, lọc ảnh, biến đổi ảnh

Nhận diện đối tượng: Phát hiện đối tượng, trích xuất đặc trưng, phân loại đối tượng

Theo dõi đối tượng: Theo dõi đối tượng trên video, ước lượng vị trí, dự báo động tương lai

Trích xuất thông tin: Trích xuất thông tin từ ảnh và video, nhận diện chữ viết tay, nhận diện ký tự trên bảng biển giao thông

1.3.3. Ứng dụng của thị giác máy tính

Trong y tế: Phát hiện ung thư, chẩn đoán bệnh, hỗ trợ phẫu thuật robot

Trong an ninh: Nhận diện khuôn mặt, phát hiện hành vi độc hại

Trong tự động hóa: Nhận diện đối tượng trong robot, xe tự hành

Nhận diện khuôn mặt và phân tích biểu cảm trên khuôn mặt

Trên đây là một số ví dụ về ứng dụng của thị giác máy tính. Tuy nhiên, danh sách này vẫn còn rất dài và liên tục mở rộng khi công nghệ phát triển.

CHƯƠNG 2: MẠNG HỒI QUY TRUY HỒI - RECURRENT NEURAL NETWORK VÀ MẠNG TÍCH CHẬP - CONVOLUTION NEURAL NETWORK

2.1. Mạng hồi quy truy hồi (Recurrent Neural Network)

Mạng thần kinh hồi quy (Recurrent Neural Network) hay RNN là một họ của mạng nơron dùng để xử lý dạng dữ liệu dạng chuỗi, đặc biệt là chuỗi dài, mạng thần kinh hồi quy đặc biệt hiệu quả với loại dữ liệu của đề tài này.

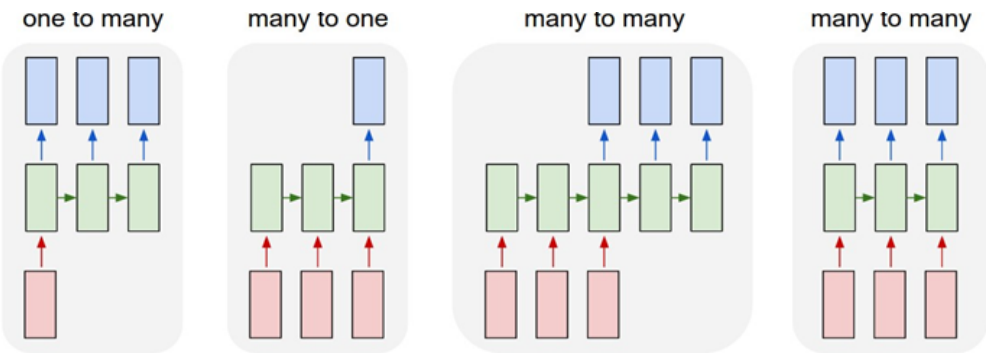
Để phát triển mạng truy hồi từ mạng lan truyền thuật đa tầng, chúng ta sẽ tận dụng ý tưởng là dùng chung tham số (parameter sharing) giữa các phần khác nhau của mô hình. Dùng chung tham số giúp ta có thể mở rộng và áp dụng mô hình trên các mẫu huấn luyện có kích thước khác nhau (trong phạm vị của đề tài là các văn bản có độ dài khác khác nhau). Nếu sử dụng bộ tham số riêng biệt tại mỗi thời điểm mô hình, mô hình sẽ không thể khái quát hóa để xử lý các chuỗi có độ dài chưa từng xuất hiện trong tập huấn luyện [3].

Sự chia sẻ này là quan trọng khi một đoạn thông tin nào đó có thể xuất hiện tại nhiều vị trong chuỗi. Xét câu: ”tôi đã đi tới Nepal trong năm 2009” và “trong năm 2009, tôi đã đi tới Nepal”. Nếu ta giao nhiệm vụ cho mô hình học máy đọc từng câu và trích xuất ra thời điểm người đó đi Nepal, thì mô hình nhận định được năm 2009 là câu trả lời. Dù nó xuất hiện ở vị trí thứ 8 hoặc 3 trong câu. Giả xử ta huấn luyện mạng lan truyền thuận để xử lí các câu có độ dài nhất định. Mạng lan truyền kết nối đầy đủ sẽ có các tham số riêng biệt cho mỗi đặc trưng đầu vào, do đó nó cần học hết các quy tắc ngôn ngữ một cách riêng biệt cho từng vị trí trong câu.

2.2. Mô hình bài toán mạng thần kinh hồi quy (mô hình mạng thần kinh hồi quy, loss function)

2.2.1. Dữ liệu dạng chuỗi

Dữ liệu văn bản có thể xem là một dữ liệu có thứ tự hay tuần tự được gọi là sequence (chuỗi), time-series (dữ liệu theo thời gian). Trong bài toán phân loại văn bản thì vị trí các từ và sự sắp xếp cực kỳ quan trọng đến nghĩa của câu và dữ liệu đầu vào được gọi là sequence data. Trong bài toán xử lý ngôn ngữ tự nhiên (NLP) thì không thể xử lý cả câu được và người ta tách ra từng từ làm mảnh nhỏ (mảnh nhỏ có thể là từ, chữ cái, cụm câu …).



Hình 4. Phân loại dạng bài toán dữ liệu chuỗi [4]

Mỗi khối là một vectơ và mũi tên là biểu diễn cho một phép biến đổi (phép nhân ma trận …). Dữ liệu đầu vào là màu đỏ, dữ liệu đầu ra là xanh dương, xanh lá là vectơ trạng thái của mạng thần kinh hồi quy. Từ trái qua phải ta phân loại bài toán mạng thần kinh hồi quy thành 4 dạng [3]:

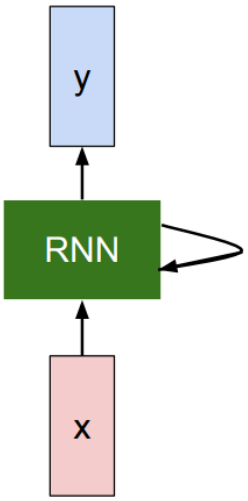
(1) Dữ liệu đầu ra là dữ liệu có thứ tự (các dạng bài chú thích ảnh image captioning).

(2) Dữ liệu đầu vào là dữ liệu có thứ tự (các dạng bài phân loại văn bản sentiment analysis).

(3) Dữ liệu đầu vào và đầu ra là dữ liệu có thứ tự (các dạng bài dịch ngôn ngữ).

(4) Đồng bộ hóa dữ liệu tuần tự đầu vào và ra (các dạng phân loại video và gán nhãn văn bản cho mỗi khung hình).

2.2.2. Hàm mất mát(Loss function) và hàm kích hoạt(Activation function)



Hình 5: Sơ đồ mô hình mạng thần kinh hồi quy[5]

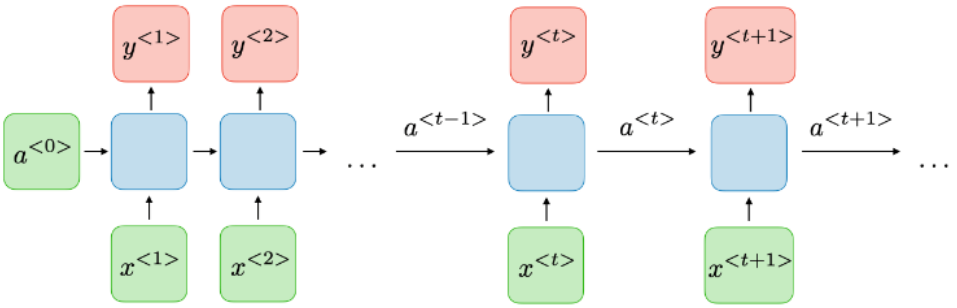
Chúng ta có thể xử lý một chuỗi các vectơ x bằng cách áp dụng công thức truy hồi ở mọi bước:

$$h_t = f_w(h_{t-1}, x_t) \text{ [10]}$$

Với h_t là trạng thái mới, f_w là một hàm với tham số w, h_{t-1} là trạng thái cũ, x_t là vector đầu vào tại mỗi bước.

mạng thần kinh hồi quy tạo ra đầu ra: Chúng ta có thể xử lý một chuỗi các vectơ x bằng cách áp dụng công thức truy hồi ở mọi bước:

$$y_t = f_{W_{hy}}(h_1)[11]$$



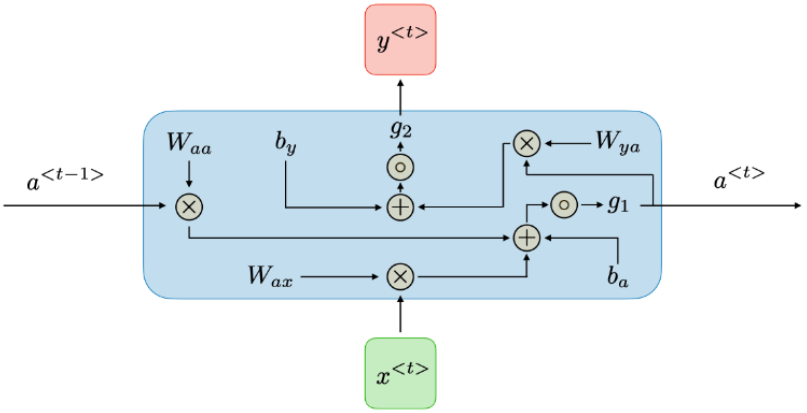
Hình 6: Kiến trúc của mạng thần kinh hồi quy[6]

Với mỗi bước lặp t, hàm kích hoạt $a^{<t>}$ và đầu ra $y^{<t>}$ được trình bày như sau:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \text{ [12]}$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)[13]$$

Với W_{ax}, W_{aa}, W_{ya}, b_a, b_y là các hệ số mà được chia sẻ tạm thời và g_1, g_2 là hai hàm kích hoạt.



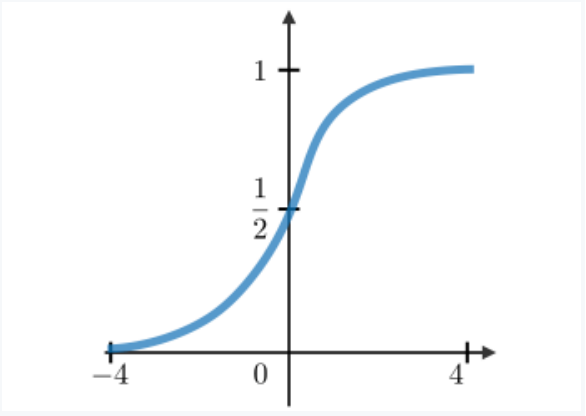
Hình 7: Sơ đồ mô tả hàm kích hoạt[7]

Trong trường hợp mạng nơ-ron hồi quy, hàm mất mát L của mọi bước thời gian được xác định dựa trên tổn thất ở mọi bước thời gian như sau:

$$L(\hat{y}, y) = \sum_{t=1}^{T_y} L(\hat{y}^{<t>}, y^{<t>})$$

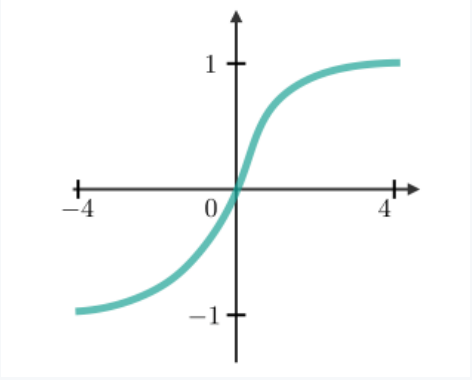
Các hàm kích hoạt thường sử dụng là:

Sigmoid: $g(z) = \frac{1}{1+e^{-z}}$



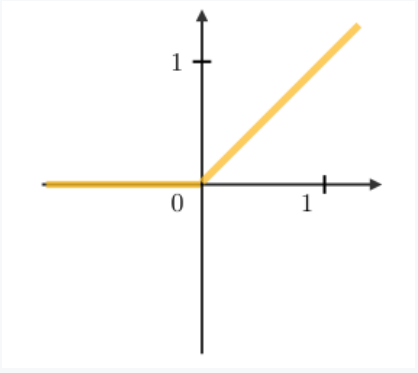
Hình 8: Đồ thị hàm sigmod[8]

Tanh: $g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$



Hình 9: Đồ thị hàm tanh [9]

RELU: $g(z) = max(0, z)$



Hình 10: Đồ thị hàm RELU[10]

Ta cần **long term memory** (bộ nhớ dài hạn) điều mà mạng thần kinh hồi quy không làm được nên ta cần một mô hình mới để giải quyết vấn đề này.

2.2.3. Lan truyền ngược(Back-Propagation)

Kỹ thuật này sẽ cập nhập lại các trọng số của nơron trong mạng theo tỷ lệ với mức độ đóng góp của lỗi chung cho toàn bộ. Lan truyền ngược là giải thuật cốt lõi giúp cho các mô hình học sâu có thể dễ dàng thực thi tính toán ngược. Với các mạng nơron hiện đại, nhờ giải thuật này mà thuật toán tối ưu với đạo hàm.

Thuật toán bao gồm 2 giai đoạn:

Giai đoạn 1: Lan truyền

Lan truyền xuôi: tính toán thuận của một đầu vào của mô hình huấn luyện thông qua mạng nơron để tạo ra các kích hoạt đầu ra của lan truyền này.

Lan truyền ngược: truyền ngược các kích hoạt đầu ra của lan truyền thông qua mạng lưới nơron sử dụng mục tiêu huấn luyện mô hình để tạo ra các delta (sai lệch giữa giá trị mục tiêu và giá trị đầu ra thực tế) và tất cả đầu ra và các nơron ẩn.

Giai đoạn 2: Cập nhật trọng số

Nhân các delta đầu ra và kích hoạt đầu vào để có được gradient của trọng số của nó.

Trừ một tỷ lệ (tỷ lệ phần trăm) từ gradient của trọng số. Tỷ lệ này (tỷ lệ phần trăm) ảnh hưởng đến tốc độ và chất lượng học. Tỷ lệ này càng lớn, thì tốc độ huấn luyện nơron càng nhanh; tỷ lệ này càng thấp, thì việc huấn luyện càng chậm. Dấu của gradient của một trọng số chỉ ra chỗ mà sai số đang gia tăng, đây là lý do tại sao trọng số phải được cập nhật theo hướng ngược lại.

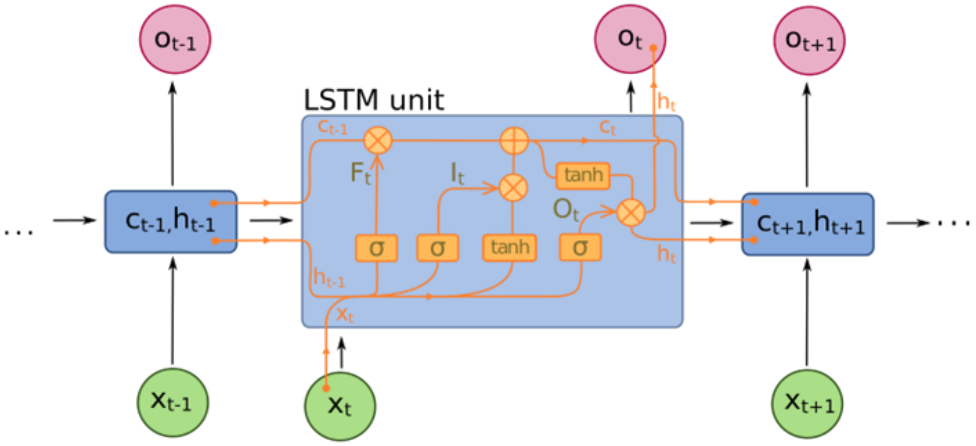
Lặp lại giai đoạn 1 và 2 cho đến khi trọng số của mạng nơron chấp nhận được

Lan truyền ngược được thực hiện tại mỗi thời điểm. Tại dấu thời gian T, đạo hàm của tổn thất L đối với ma trận trọng số W được biểu thị như sau:

$$\frac{\partial L^{(T)}}{\partial W} = \sum_{t=1}^{T} \frac{\partial L^{(T)}}{\partial W}_{(t)} \quad [14]$$

2.2.4. Long Short-Term Memory (LSTM)

Long Short-Term Memory (hay LSTM), là một họ của mạng thần kinh hồi quy trong học sâu, được tạo ra với mục đích xử lý dữ liệu chuỗi và khắc phục được vấn đề vanishing gradient và short-term memory. Ta có mô hình Long Short-Term Memory như sau:



Hình 10. Mô hình Long Short-Term Memory (nguồn: Internet)

Tại state thứ t của mô hình Long Short-Term Memory:

Đầu ra: c_t, h_t ta gọi c là cell state, h là hidden state

Đầu vào: c_{t-1}, h_{t-1} , x_t . Trong đó x_t là đầu vào ở state thứ t của mô hình. c_{t-1}, h_{t-1}là output của layer trước. h đóng vai trò khá giống như s ở mạng thần kinh hồi quy, trong khi c là điểm mới của Long Short-Term Memory

với f_t, i_t, o_t lần lượt là forget gate, input gate và output gate và được xác định:

Forget gate: $f_t = \sigma(U_f * x_t + W_f * h_{t-1} + b_f)$

Input gate: $i_t = \sigma(U_i * x_t + W_f * h_{t-1} + b_i)$

Output gate: $o_t = \sigma(U_o * x_t + W_f * h_{t-1} + b_o)$

Với:

$0 < f_t, i_t, o_t < 1$

b_f, b_i, b_o là các hệ số bias; hệ số W, U giống trong mạng thần kinh hồi quy

$\widehat{c} = tanh(U_c * x_t + W_c * h_{t-1} + b_c)$, bước này giống hệt như tính s_t trong mạng thần kinh hồi quy $c_t = f_c * c_{t-1} + i_t * \widehat{c}$, forget gate quyết định xem cần lấy bao nhiêu từ cell state trước và input gate sẽ quyết định lấy bao nhiêu từ input của state và hidden layer của layer trước.

$h_t = o_t * tanh(c_t)$, output gate quyết định xem cần lấy bao nhiêu từ cell state để trở thành output của hidden state. Ngoài ra h_t cũng được dùng để tính ra output y_t cho state t.

h_t và $\widehat{c_t}$ giống với mạng thần kinh hồi quy, nên mô hình có short term memory. Trong khi đó $c_t$giống như một băng chuyền ở trên mô hình mạng thần kinh hồi quy vậy, thông tin nào cần quan trọng và dùng ở sau sẽ được gửi vào và dùng khi cần ⇒ có thể mang thông tin từ đi xa (long term memory). Do đó mô hình Long Short-Term Memory có cả short term memory và long term memory.

Khi lan truyền ngược cho Long Short-Term Memory, ta cũng đã xác định được thành phần chính gây vanishing gradient trong mạng thần kinh hồi quy là , trong đó s_t, $W < 1$. Tương tự trong Long Short-Term Memory ta quan tâm đến , do $0 < f_t < 11$ nên về cơ bản thì Long Short-Term Memory vẫn bị vanishing gradient nhưng bị ít hơn so với mạng thần kinh hồi quy. Hơn thế nữa, khi mang thông tin trên cell state thì ít khi cần phải quên giá trị cell cũ, nên $f_t \approx 1$ và tránh được vanishing gradient. Do đó Long Short-Term Memory được dùng phổ biến hơn mạng thần kinh hồi quy cho các toán thông tin dạng chuỗi.

2.4. Mạng tích chập (tích chập)

Mạng tích chập (Convolutional Neural Network) là một loại mạng neural được thiết kế đặc biệt để xử lý dữ liệu có cấu trúc ruộng ảnh và video. Nó là một trong những kiến trúc quan trọng nhất trong lĩnh vực thị giác máy tính và xử lý ảnh.

Các mạng tích chập được lấy cảm hứng từ cách xử lý thông tin của hệ thống thị giác của con người. Kiến trúc của chúng bao gồm các lớp đặc biệt như lớp tích chập (convolutional layer), lớp pooling và lớp kết nối đầy đủ (fully connected layer).

Lớp tích chập là lõi của một tích chập. Nó sử dụng các bộ lọc (filters) để thực hiện phép tích chập trên dữ liệu đầu vào. Các bộ lọc này giúp tìm ra các đặc trưng quan trọng trong ảnh, như cạnh, góc, hoặc các hình dạng đặc biệt. Kết quả của phép tích chập là các bản đồ đặc trưng (feature maps) chứa thông tin về những đặc trưng này.

Lớp pooling được sử dụng để giảm kích thước của bản đồ đặc trưng. Phương pháp phổ biến nhất là pooling theo phạm vi (max pooling), trong đó chỉ giữ lại giá trị lớn nhất trong mỗi vùng của bản đồ đặc trưng. Điều này giúp giảm số lượng tham số và tính toán của mạng, đồng thời giữ lại thông tin quan trọng nhất.

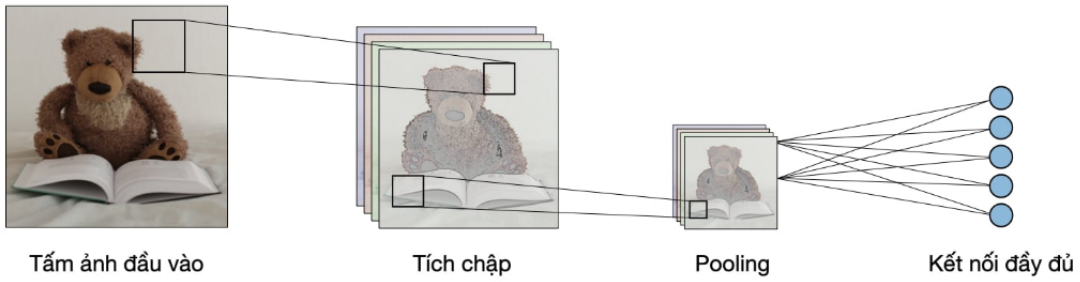
Cuối cùng, lớp kết nối đầy đủ được sử dụng để phân loại hoặc dự đoán. Nó nhận các đặc trưng đã được trích xuất từ lớp tích chập và pooling và áp dụng các phép biến đổi tuyến tính để tạo ra đầu ra cuối cùng.

Mạng tích chập được huấn luyện thông qua quá trình lan truyền ngược (backpropagation) để điều chỉnh các trọng số của nó sao cho đầu ra dự đoán của mạng gần với kết quả mong muốn. Các phương pháp tối ưu hóa như stochastic gradient descent (SGD) thường được sử dụng trong quá trình này.

Mạng tích chập đã đạt được những thành công ấn tượng trong nhiều ứng dụng, bao gồm nhận dạng ảnh, phân loại đối tượng, nhận dạng khuôn mặt, phân tích cảm xúc và xe tự lái. Kiến trúc mạng tích chập được phát triển và cải tiến liên tục, đi kèm với sự phát triển của các phương pháp và công nghệ mới trong lĩnh vực thị giác máy tính và học sâu.

*Kiến trúc của tích chập:

Mạng thần kinh chuyển đổi bao gồm nhiều lớp như lớp đầu vào, lớp chuyển đổi, lớp tổng hợp và các lớp được kết nối đầy đủ. Hình 11 là một mô tả về sơ đồ mô hình mạng tích chập



Hình 11: Sơ đồ mô hình mạng tích chập(Nguồn: Đại học Stanford)[11]

2.4.1. ResnetModel

ResNet (Residual Neural Network) là một kiến trúc mạng tích chập đặc biệt với việc sử dụng khối dư (residual blocks) để xây dựng mạng. Kiến trúc này đã giúp giải quyết vấn đề về hiện tượng đường cong học biến mất (vanishing gradients) trong các mạng sâu.

ResNet (Residual Neural Network) là một kiến trúc mạng tích chập được giới thiệu bởi Kaiming He và đồng nghiệp vào năm 2015. Nó đã đạt được nhiều thành công đáng kể trong các cuộc thi về nhận dạng ảnh và phân loại.

Ý tưởng chính của ResNet là sử dụng các residual blocks (khối dư) để xây dựng kiến trúc mạng. Mỗi khối dư bao gồm một chuỗi các lớp tích chập và lớp kết nối đầy đủ, với việc thêm một đường kết nối "skip" (đường bỏ qua) để tránh hiện tượng đường cong học biến mất (vanishing gradients) và cho phép thông tin truyền từ lớp này sang lớp khác một cách dễ dàng hơn.

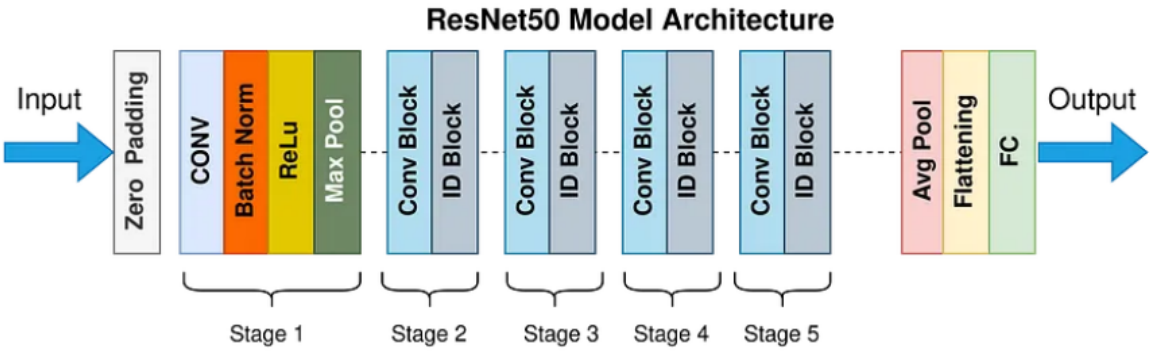
Một khối dư (residual block) trong ResNet bao gồm một chuỗi các lớp tích chập và lớp kết nối đầy đủ. Trong mỗi khối dư, đầu vào của khối được truyền qua các lớp tích chập để trích xuất đặc trưng. Tuy nhiên, thay vì định nghĩa trực tiếp một hàm ánh xạ từ đầu vào đến đầu ra, ResNet định nghĩa một hàm residual (hàm còn lại) F(x).

Cách tiếp cận của ResNet là sử dụng kết nối "skip" (đường bỏ qua) để truyền thông tin trực tiếp từ đầu vào sang đầu ra của mỗi khối dư. Điều này được thực hiện bằng cách thêm đầu vào ban đầu (x) vào đầu ra của hàm residual (F(x)). Kết quả là đầu ra của khối dư được tính bằng F(x) + x.

Việc thêm đường kết nối "skip" nhằm tránh hiện tượng đường cong học biến mất (vanishing gradients), tức là sự suy giảm đáng kể của độ dốc khi đường ngược (backpropagation) được áp dụng để điều chỉnh các trọng số của mạng. Đường kết nối "skip" cho phép thông tin truyền từ lớp này sang lớp khác một cách dễ dàng hơn, giúp mạng học cụ thể về sự khác biệt (residual) giữa đầu vào và đầu ra.

ResNet thường được xây dựng với nhiều khối dư, và số lượng khối dư sẽ phụ thuộc vào độ phức tạp và độ sâu mạng mong muốn. Các phiên bản phổ biến của ResNet bao gồm ResNet-18, ResNet-34, ResNet-50, ResNet-101 và ResNet-152, với số lượng khối dư tương ứng là 18, 34, 50, 101 và 152.

ResNet đã chứng minh được hiệu suất tốt trong nhiều tác vụ như nhận dạng đối tượng, phân loại ảnh và phát hiện vật thể. Nhờ việc sử dụng các khối dư và đường kết nối "skip", ResNet giúp giảm hiện tượng mất mát thông tin và cho phép xây dựng những mạng tích chập sâu hơn mà vẫn đạt được kết quả tốt.



Hình 12: Kiến trúc mô hình Resnet50[12]

CHƯƠNG 3: TỔNG QUAN VỀ BÀI TOÁN TẠO CHÚ THÍCH HÌNH ẢNH (IMAGE CAPTIONING)

3.1. Tổng quan về bài toán tạo chú thích cho hình ảnh

3.1.1. Khái niệm

Bài toán image captioning là một trong những bài toán được quan tâm nhiều nhất trong thị giác máy tính, với mục tiêu tạo ra các mô hình và thuật toán cho máy tính có khả năng tạo ra mô tả (caption) tự động cho một hình ảnh đầu vào. Với sự phát triển của trí tuệ nhân tạo và deep learning, các mô hình image captioning đạt được kết quả khá ấn tượng, và có thể được ứng dụng trong nhiều lĩnh vực như tự động tạo chú thích cho ảnh, xử lý ảnh y tế và dược phẩm, và tạo ra các trò chơi điện tử đa dạng.

3.1.2. Lịch sử phát triển của bài toán tạo chú thích hình ảnh

Lịch sử phát triển của bài toán này bắt đầu vào năm 2014, khi xuất hiện bài báo "Deep Visual-Semantic Alignments for Generating Image Descriptions" của Vinyals và các đồng nghiệp tại Google. Bài báo này đề xuất một mô hình neural network kết hợp giữa mạng nơ-ron tích chập (tích chập) và mạng thần kinh hồi quy (RNN) để tạo ra chú thích cho hình ảnh.

Sau đó, nhiều nghiên cứu đã được thực hiện với các cải tiến đáng kể về kiến trúc và độ chính xác. Năm 2015, Karpathy và Li của đại học Stanford đã đề xuất mô hình "Deep Visual-Semantic Alignments for Generating Image Descriptions with Human Parity" với độ chính xác xấp xỉ ngang ngữ nghĩa của con người. Năm 2016, Google phát hành bộ dữ liệu "Google Conceptual Captions" để đào tạo các mô hình image captioning với độ phức tạp cao hơn.

Tóm lại ,bài toán tạo chú thích hình ảnh đã có sự phát triển đáng kể trong những năm qua và đang là một trong những lĩnh vực nghiên cứu được quan tâm rất nhiều trong cộng đồng trí tuệ nhân tạo.

3.2. Các phương pháp tiếp truyền thống và các mô hình deep learning được áp dụng cho bài toán

Các phương pháp tiếp truyền thống: sử dụng các mô hình xác suất như Hidden Markov Models (HMM) và Maximum Entropy Markov Models (MEMM), cùng với các phương pháp tìm kiếm như Beam Search và Viterbi Algorithm để đưa ra chú thích

cho hình ảnh. Tuy nhiên, các phương pháp này thường không đủ chính xác và phải đối mặt với vấn đề của việc xử lý ngôn ngữ tự nhiên.

Các mô hình deep learning đầu tiên được áp dụng cho bài toán image captioning: mô hình Recurrent Neural Networks (RNN) và Long Short-Term Memory (Long Short-Term Memory), cho phép mô hình học cách mô tả hình ảnh dựa trên thông tin đã học được từ các văn bản mô tả. Tuy nhiên, các mô hình này còn đối mặt với vấn đề đưa ra các câu mô tả ngắn và không đủ chính xác.

Các mô hình deep learning được sử dụng phổ biến cho bài toán image captioning bao gồm mô hình Convolutional Neural Networks (tích chập) và Transformer-based models nhưng vẫn đang tiếp tục được nghiên cứu và cải tiến để đạt được kết quả tốt hơn.

Input:

- Hình ảnh: Đây là input chính cho bài toán Image Captioning. Hình ảnh có thể là bất kỳ hình ảnh nào, ví dụ như một bức ảnh chứa các đối tượng hoặc một cảnh quan.
- Output: Mô tả văn bản: Output của bài toán Image Captioning là một câu mô tả văn bản tự động cho hình ảnh đầu vào. Câu mô tả này cung cấp thông tin về nội dung và bối cảnh của hình ảnh.

Các bước thực hiện để đi từ input ra output:

Bước 1: Tiền xử lý hình ảnh:

- Hình ảnh đầu vào được truyền qua một mạng tích chập để trích xuất các đặc trưng. Các đặc trưng này có thể được trích xuất từ các tầng tích chập cuối cùng hoặc từ các tầng trung gian của mạng tích chập.

Bước 2: Xây dựng mô hình ngôn ngữ:

- Mô hình ngôn ngữ được sử dụng để tạo ra câu mô tả từ các đặc trưng hình ảnh. Mô hình này thường dựa trên mạng thần kinh hồi quy (Recurrent Neural Network) như Long Short-Term Memory (Long Short-Term Memory) hoặc GRU (Gated Recurrent Unit). Nó nhận đầu vào là các đặc trưng hình ảnh và dự đoán từng từ hoặc phân đoạn từ của mô tả.

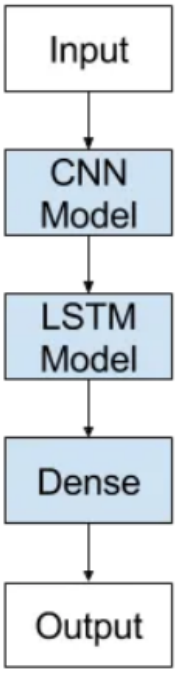
Bước 3: Huấn luyện mô hình:

- Mô hình ngôn ngữ được huấn luyện bằng cách sử dụng một tập dữ liệu gồm các cặp hình ảnh-văn bản. Quá trình huấn luyện nhằm điều chỉnh các tham số của mô hình để tối ưu hóa việc dự đoán mô tả chính xác cho các hình ảnh đầu vào.

Bước 4: Dự đoán và sinh mô tả:

- Sau khi mô hình đã được huấn luyện, nó có thể được sử dụng để dự đoán và sinh mô tả cho các hình ảnh mới. Quá trình này bắt đầu bằng việc đưa hình ảnh mới qua mạng tích chập để trích xuất đặc trưng. Sau đó, các đặc trưng được đưa vào mô hình ngôn ngữ để dự đoán từng từ hoặc phân đoạn từ của mô tả.

Bước 5: Đánh giá mô hình: Mô hình Image Captioning được đánh giá bằng cách so sánh mô tả được sinh ra với mô tả thực tế của hình ảnh. Độ chính xác và độ tương tự giữa mô tả sinh ra và mô tả thực tế được đo bằng các độ đo như BLEU (BiLingual Evaluation Understudy) hoặc METEOR (Metric for Evaluation of Translation with Explicit ORdering).



Hình 13: Cấu trúc thực hiện của bài toán mà nhóm thực hiện

3.3. Các thách thức của bài toán

Bài toán tạo chú thích hình ảnh (image captioning) đang đối mặt với một số thách thức:

+ *Độ phức tạp của dữ liệu hình ảnh*: Hình ảnh có thể chứa nhiều chi tiết khác nhau, có thể có nhiễu, góc chụp khác nhau, độ sáng khác nhau, độ phân giải khác nhau, làm cho việc phân tích và trích xuất đặc trưng từ hình ảnh trở nên khó khăn.

+ *Sự đa dạng của ngôn ngữ*: Một hình ảnh có thể được mô tả bằng nhiều cách khác nhau bởi con người. Vì vậy, việc tạo ra các chú thích hình ảnh chính xác và tự nhiên đòi hỏi mô hình phải có khả năng phân tích và hiểu được ngôn ngữ tự nhiên.

+ *Độ dài của chú thích*: Chú thích hình ảnh cần phải cung cấp đủ thông tin mô tả về hình ảnh nhưng vẫn phải ngắn gọn và dễ hiểu.

+ *Số lượng dữ liệu đào tạo*: Để đào tạo các mô hình image captioning, cần phải có một lượng lớn dữ liệu hình ảnh kết hợp với chú thích. Tuy nhiên, việc xây dựng bộ dữ liệu phức tạp và đầy đủ là một thách thức.

+ *Tốc độ và hiệu suất*: Mô hình phải có khả năng tạo ra chú thích trong thời gian thực và đáp ứng được các yêu cầu về tốc độ và hiệu suất.

+ *Độ phức tạp của mô hình:* Mô hình deep learning có thể có cấu trúc phức tạp và yêu cầu nhiều tài nguyên tính toán, đặc biệt là khi áp dụng trên các ứng dụng thực tế.

3.4. Tiền xử lý dữ liệu

Trước khi đưa dữ liệu vào mô hình, cần tiền xử lý dữ liệu bằng cách chuyển đổi hình ảnh thành một ma trận số và chuyển đổi câu chú thích thành một vector số. Việc này giúp mô hình có thể xử lý dữ liệu dễ dàng hơn.

3.5. Sử dụng mạng tích chập để trích xuất đặc trưng

Một hình ảnh đầu vào được xử lý qua một mạng tích chập để trích xuất ra các đặc trưng. Các mạng tích chập thường được huấn luyện trên các bộ dữ liệu lớn như ImageNet để học cách phân loại hình ảnh. Sau đó, các mạng tích chập này được sử dụng lại cho bài toán image captioning bằng cách sử dụng các đặc trưng được học từ những tầng cuối cùng của mạng. Các đặc trưng này sẽ được đưa vào một mạng thần

kinh hồi quy để sinh ra câu chú thích cho hình ảnh. Một số mô hình sử dụng mạng Long Short-Term Memory để học được sự phụ thuộc dài hạn giữa các từ trong câu chú thích.

3.6. Sử dụng mạng thần kinh hồi quy để sinh ra câu chú thích

Một mạng thần kinh hồi quy sẽ được sử dụng để sinh ra câu chú thích từ các đặc trưng được trích xuất từ hình ảnh. Các mạng thần kinh hồi quy thường được sử dụng là mạng Long Short-Term Memory hoặc GRU để học được sự phụ thuộc dài hạn giữa các từ trong câu chú thích. Một cách tiếp cận phổ biến cho bài toán này là sử dụng mạng thần kinh hồi quy để sinh ra từng từ trong câu chú thích. Mỗi từ được sinh ra dựa trên các từ đã sinh ra trước đó và các đặc trưng của hình ảnh. Sau đó, các từ này sẽ được sắp xếp lại để tạo ra câu chú thích cho hình ảnh.

3.7. Huấn luyện mô hình

Trước khi huấn luyện mô hình, cần có một tập dữ liệu gồm các cặp hình ảnh và câu chú thích tương ứng. Mô hình sẽ được huấn luyện để tối ưu hóa hàm mất mát giữa câu chú thích được dự đoán và câu chú thích thực tế. Việc huấn luyện mô hình có thể mất rất nhiều thời gian và tài nguyên tính toán.

3.8. Đánh giá mô hình

Mô hình được đánh giá bằng cách so sánh câu chú thích được dự đoán với câu chú thích thực tế. Các phương pháp đánh giá bao gồm độ chính xác, độ đa nghĩa, độ đo F1 và BLEU.

Độ đô BLEU:

Độ đo BLEU (BiLingual Evaluation Understudy) là một phương pháp thường được sử dụng để đánh giá chất lượng các dịch thuật tự động, đặc biệt là trong lĩnh vực xử lý ngôn ngữ tự nhiên. BLEU đo lường độ tương đồng giữa câu dịch tự động và câu tham chiếu (được tạo bởi con người). Quá trình đánh giá bắt đầu bằng việc so sánh các từ trong câu dịch tự động với các từ trong câu tham chiếu.

BLEU sử dụng n-gram để đo lường tần suất xuất hiện của các từ hoặc chuỗi từ trong câu dịch tự động và so sánh với câu tham chiếu. Mức độ khớp n-gram cao sẽ cho điểm cao hơn. Điểm số BLEU được tính toán dựa trên n-gram precision và một yếu tố rút gọn (brevity penalty) để đảm bảo rằng dịch thuật ngắn hơn không nhận điểm cao

chỉ vì số từ ít.

Điểm BLEU thường được trình bày dưới dạng phần trăm (từ 0 đến 100%), với giá trị cao hơn cho thấy sự tương đồng cao hơn giữa câu dịch tự động và câu tham chiếu. Tuy nhiên, BLEU cũng có nhược điểm. Vì nó chỉ sử dụng các độ đo ngữ cảnh cục bộ (n-gram) và không xem xét các khía cạnh ngữ nghĩa hay ngữ cảnh toàn cục của câu, nên đánh giá BLEU có thể không phản ánh chính xác chất lượng của câu dịch tự động một cách toàn diện. BLEU là một độ đo phổ biến để đánh giá chất lượng dịch thuật tự động, nhưng nên được sử dụng cùng với các phương pháp và độ đo khác để có cái nhìn đa chiều hơn về chất lượng dịch thuật.

BP (brevity penalty) là hệ số phạt viết tắt để giải quyết vấn đề khi các chuỗi dự đoán quá ngắn so với chuỗi tham chiếu. BP được tính bằng cách so sánh độ dài của chuỗi ngắn nhất trong các chuỗi dự đoán với độ dài của các chuỗi tham chiếu. Nếu chuỗi dự đoán ngắn hơn chuỗi tham chiếu, BP sẽ giảm giá trị BLEU.

N là số lượng từ trong các chuỗi dự đoán.

w_t là trọng số của từ thứ t, thường được gán bằng $\frac{1}{N}$.

precision là độ chính xác của từ thứ i trong các chuỗi dự đoán. Precision được tính bằng số lượng từ thứ i trong các chuỗi dự đoán giống với từ thứ i trong các chuỗi tham chiếu, chia cho tổng số lượng từ thứ i trong các chuỗi dự đoán.

Các giá trị BLEU có giá trị từ 0 đến 1, với giá trị càng cao thì mô hình đưa ra dự đoán càng chính xác.

Công thức:

$$Unigram\ Precision\ P\ = \frac{m}{w_t}$$

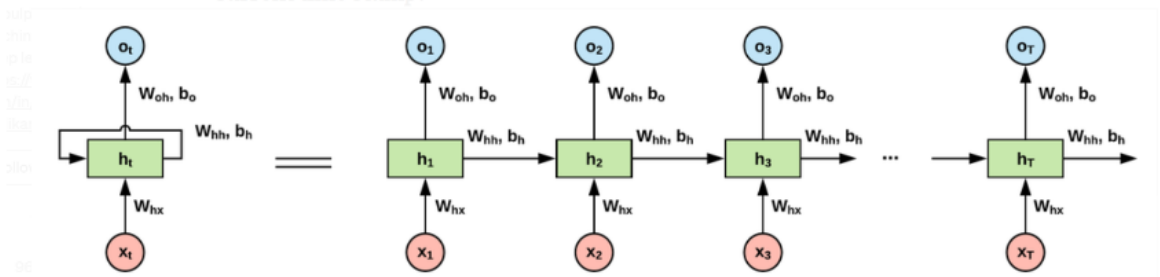
$$Brevity\ penalty\ P\ =\ 1\ if\ c\ >\ r$$

$$Brevity\ penalty\ P\ =\ e^{(1-\frac{r}{c})}\ if\ c\ \leq r$$

$$BLEU\ =\ p\ *\ e^{\sum_{1}^{N}(\frac{1}{N}*logPn)}$$

3.8. Hạn chế của mô hình thần kinh hồi quy

- Training thần kinh hồi quy



Hình 14: Sơ đồ train mô hình thần kinh hồi quy[13]

Hình 13 cho thấy một khối thần kinh hồi quy điển hình trông như thế nào. Như chúng ta có thể thấy, các thần kinh hồi quy cũng lấy đầu ra của nút trước đó làm đầu vào ở trạng thái hiện tại. Chà, nó giúp hiểu đúng ngữ cảnh nhưng lại thất bại ở một khía cạnh. Cực tiểu hóa hàm mất mát. Dưới đây là các chức năng kích hoạt thường được sử dụng với thần kinh hồi quy. Vấn đề với các hàm kích hoạt này là bất cứ khi nào chúng được sử dụng trong huấn luyện tuần tự, trọng số (hoặc độ dốc) làm cho quá trình trở nên khó khăn hơn một chút.

- Vấn đề về vanishing or exploding gradient

Các vấn đề về độ dốc biến mất và/hoặc bùng nổ thường xảy ra đối với thần kinh hồi quy. Động lực đằng sau lý do tại sao chúng xảy ra là rất khó để nắm bắt các điều kiện đường dài do góc độ nhân có thể giảm/mở rộng đáng kể về số lượng lớp. Vì vậy, nếu tất cả các trình tự trở nên quá dài, mô hình có thể huấn luyện với trọng số rỗng (tức là không huấn luyện) hoặc trọng số phát nổ.

- Exploding gradients

Quá trình đào tạo bất kỳ thần kinh hồi quy chưa mở nào được thực hiện qua nhiều bước thời gian, trong đó chúng tôi tính toán độ dốc lỗi là tổng của tất cả các lỗi độ dốc qua các dấu thời gian. Do đó, thuật toán còn được gọi là lan truyền ngược theo thời gian (BPTT). Do ứng dụng quy tắc dây chuyền trong khi tính toán độ dốc lỗi, sự chi phối của số hạng nhân tăng theo thời gian và do đó độ dốc có xu hướng bùng nổ hoặc biến mất. Nếu giá trị riêng lớn nhất nhỏ hơn 1, độ dốc sẽ biến mất. Nếu giá trị riêng lớn nhất lớn hơn 1, độ dốc sẽ bùng nổ.

Vấn đề bùng nổ gradient có thể được giải quyết bằng cách sử dụng gradient clipping. Như tên gợi ý, các chuyển màu được cắt bớt khi chúng đạt đến ngưỡng được

xác định trước. Những vấn đề biến mất độ dốc vẫn còn tồn tại. Sau đó, nó đã được giải quyết với sự ra đời của các mạng Long Short-Term Memory. Có các phương pháp khác để giải quyết vấn đề thao tác gradient phụ thuộc giá trị riêng này. Ví dụ: hình phạt L1 và L2 của trọng số và độ dốc lặp lại. Đó là một thủ thuật/kỹ thuật hack đơn giản mà qua đó chúng ta có thể chuẩn hóa hoặc loại bỏ các giá trị riêng khi nó tăng cao trong giá trị nhân.

- Thần kinh hồi quy không thể xếp chồng lên nhau

Vấn đề số một khi đề cập đến việc song song hóa các khóa đào tạo trong mạng thần kinh hồi quy hoặc một lớp đào tạo đơn giản là do đặc điểm cơ bản của mạng thần kinh hồi quy, tức là chúng được kết nối với nhau. Điều này có nghĩa là mạng thần kinh hồi quys yêu cầu đầu ra của nút trước đó để thực hiện tính toán trên nút hiện tại. Do kết nối này, mạng thần kinh hồi quy không phù hợp để song song hóa hoặc xếp chồng lên nhau với các mô hình khác. Chi phí tính toán tổng thể tiếp tục không bao giờ có thể được chứng minh bằng bất kỳ độ chính xác nào.

- Quy trình đào tạo chậm và phức tạp

Một trong những vấn đề cơ bản với mạng thần kinh hồi quy là chúng có tính lặp lại. Có nghĩa là họ sẽ mất rất nhiều thời gian cho việc đào tạo. Tốc độ đào tạo tổng thể của mạng thần kinh hồi quy khá thấp so với các mạng feedforward. Thứ hai, do mạng thần kinh hồi quy cần hiệu chỉnh các đầu ra trước đó cũng như các đầu vào hiện tại thành chức năng thay đổi trạng thái trên mỗi nút nên khá khó thực hiện. Sự phức tạp của đào tạo đôi khi khiến việc tùy chỉnh đào tạo mạng thần kinh hồi quy trở nên khó khăn hơn.

- Khó xử lý các chuỗi dài hơn

Rất khó để huấn luyện mạng thần kinh hồi quy trên các chuỗi quá dài, đặc biệt là khi sử dụng kích hoạt ReLU hoặc tanh. Đây là một lý do khác để giới thiệu các mạng dựa trên GRU.

3.9. Hạn chế của Long Short-Term Memory

- Đầu tiên, chúng phức tạp hơn mạng thần kinh hồi quy truyền thống và yêu cầu nhiều dữ liệu huấn luyện hơn để học hiệu quả.

- Thứ hai, chúng không phù hợp cho các online learning task, chẳng hạn như các nhiệm vụ dự đoán hoặc phân loại mà dữ liệu đầu vào không phải là một chuỗi. Thứ ba, Long Short-Term Memory có thể chậm đào tạo trên các tập dữ liệu lớn. Điều này là do thực tế là ta phải tìm hiểu các tham số của các ô Long Short-Term Memory, điều này có thể tốn nhiều công sức tính toán.
- Cuối cùng, Long Short-Term Memory có thể không phù hợp với mọi loại dữ liệu. Ví dụ: chúng có thể không hoạt động tốt với dữ liệu phi tuyến tính cao hoặc dữ liệu có nhiều nhiễu.

CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT QUẢ ĐẠT ĐƯỢC

4.1.Tập dữ liệu

Bộ dữ liệu Flickr8k được tạo ra bởi một nhóm nghiên cứu trong lĩnh vực xử lý ngôn ngữ tự nhiên và thị giác máy tính. Nghiên cứu này được công bố vào năm 2011 và có tựa đề "Building a Large-Scale Dataset for Image Annotation, Captions and Retrieval" (Xây dựng một tập dữ liệu quy mô lớn cho việc gắn nhãn hình ảnh, mô tả và tìm kiếm). Nhóm nghiên cứu đã tải lên Flickr8k bộ dữ liệu từ trang web chia sẻ ảnh Flickr. Họ đã chọn một tập hợp các hình ảnh đa dạng về nội dung và thu thập các mô tả ngắn gọn cho mỗi hình ảnh từ cộng đồng người dùng Flickr. Mục đích của bộ dữ liệu Flickr8k là cung cấp một tập dữ liệu đa dạng và phong phú để nghiên cứu và phát triển các mô hình xử lý ngôn ngữ tự nhiên và hình ảnh. Nó đã trở thành một tài nguyên quan trọng và được sử dụng rộng rãi trong cộng đồng nghiên cứu trong nhiều năm qua.

Tập dữ liệu Flickr8k là một bộ dữ liệu được sử dụng phổ biến trong lĩnh vực xử lý ngôn ngữ tự nhiên và thị giác máy tính. Nó được sử dụng để thực hiện các nhiệm vụ như mô hình hóa hình ảnh và mô tả hình ảnh.

Bộ dữ liệu Flickr8k bao gồm khoảng 8,000 hình ảnh từ trang web chia sẻ ảnh Flickr. Mỗi hình ảnh được kèm theo năm mô tả ngắn gọn. Do đó, tổng cộng có khoảng 40,000 mô tả cho các hình ảnh trong tập dữ liệu.

Các mô tả trong Flickr8k thường được viết bằng tiếng Anh và có độ dài từ 10 đến 20 từ. Đây là tập dữ liệu phổ biến trong lĩnh vực xử lý ngôn ngữ tự nhiên và đã được sử dụng rộng rãi trong nhiều nghiên cứu và công trình liên quan đến xử lý ngôn ngữ và hình ảnh. Tập dữ liệu Flickr8k cung cấp một nguồn tài nguyên quan trọng để phát triển và đánh giá các mô hình và thuật toán xử lý hình ảnh và ngôn ngữ tự nhiên.

4.2. Tiền xử lý dữ liệu

T**iền xử lý hình ảnh bao gồm các bước sau**:

Resize hình ảnh: Đầu tiên, hình ảnh đầu vào được điều chỉnh kích thước để có kích thước cố định phù hợp với mô hình. Việc này giúp đảm bảo rằng tất cả các hình ảnh có cùng kích thước và đồng nhất về đầu vào cho mô hình.

Trích xuất đặc trưng từ hình ảnh: Hình ảnh đã được resize được đưa qua mạng tích chập trước để trích xuất các đặc trưng. Mạng tích chập như VGG16 hoặc ResNet

thường được sử dụng. Các tầng tích chập cuối cùng của mạng tích chập được sử dụng để trích xuất các đặc trưng tầng cao từ hình ảnh.

Sử dụng pooling để giảm kích thước: Các đặc trưng tầng cao trích xuất từ mạng tích chập có kích thước lớn. Vì vậy, một bước pooling thường được áp dụng để giảm kích thước của đặc trưng. Max pooling hoặc average pooling thường được sử dụng để giữ lại thông tin quan trọng trong đặc trưng và giảm chiều dữ liệu.

Chuẩn hóa đặc trưng: Sau khi giảm kích thước, đặc trưng từ bước pooling được chuẩn hóa. Chuẩn hóa đặc trưng thường bao gồm việc chuyển đổi giá trị về khoảng [0, 1] hoặc [-1, 1]. Điều này giúp tối ưu quá trình huấn luyện và đảm bảo rằng các giá trị đặc trưng có cùng phạm vi.

Padding: Đôi khi, để đảm bảo rằng tất cả các hình ảnh có cùng kích thước đầu vào, các hình ảnh có kích thước nhỏ hơn được lấp đầy (padding) để có kích thước bằng nhau. Việc padding giúp duy trì sự đồng nhất trong đầu vào và đảm bảo rằng các hình ảnh có độ dài và chiều rộng như nhau.

Tiền xử lý ngôn ngữ bao gồm các bước sau:

Chuẩn hóa văn bản: Đầu tiên, các văn bản mô tả được tiền xử lý bằng cách loại bỏ các ký tự không cần thiết, dấu câu hoặc ký tự đặc biệt. Các văn bản cũng thường được chuyển thành chữ thường để đảm bảo sự thống nhất và giảm độ phức tạp của dữ liệu.

Xây dựng từ điển: Từ điển (vocabulary) được xây dựng từ các từ xuất hiện trong tập dữ liệu mô tả. Từ điển chứa tất cả các từ và có thể bao gồm các ký tự đặc biệt như startseq và endseq. Mỗi từ trong từ điển được gắn một chỉ số duy nhất.

Chuẩn bị dữ liệu đầu vào cho mô hình: Văn bản mô tả cần được chuẩn bị thành một đầu vào phù hợp cho mô hình ngôn ngữ. Mỗi câu mô tả được chia thành các từ riêng biệt. Mỗi từ được mã hóa thành một chỉ số tương ứng trong từ điển. Điều này giúp biểu diễn văn bản dưới dạng các vector số hóa có thể được đưa vào mô hình.

Padding và định dạng đầu vào: Đối với các câu mô tả có độ dài khác nhau, ta cần chuẩn bị đầu vào có cùng độ dài để đưa vào mô hình. Thông thường, ta sẽ thêm

padding vào các câu mô tả ngắn hơn để có độ dài như nhau. Đồng thời, các chuỗi được biểu diễn dưới dạng ma trận với các giá trị chỉ số từ trong từ điển.

Mã hoá đầu ra: Mô hình ngôn ngữ thường sử dụng mã hoá one-hot hoặc mã hoá nhúng (embedding) để biểu diễn các từ đầu ra. Mã hoá one-hot tạo ra một vector có độ dài bằng số từ trong từ điển, trong đó chỉ có một giá trị 1 tại vị trí tương ứng với từ cần dự đoán. Mã hoá nhúng sử dụng một ma trận nhúng để biểu diễn từng từ thành một vector dày đặc

4.3. Ứng dụng mô hình Resnet và VGG16 để trích xuất đặc trưng

Sau khi tiền xử lý:

Xây dựng mạng tích chập: Mạng tích chập được xây dựng để trích xuất đặc trưng từ hình ảnh. Các kiến trúc tích chập như VGG16, ResNet. Một số tầng tích chập cuối cùng của mạng tích chập được chọn để lấy các đặc trưng tầng cao. Tầng này có khả năng trích xuất các đặc trưng trừu tượng và có thể chứa thông tin quan trọng về hình ảnh.

Đưa hình ảnh qua mạng tích chập: Hình ảnh được đưa qua mạng tích chập để trích xuất đặc trưng. Quá trình này là một quá trình feed-forward, trong đó hình ảnh được đưa qua các tầng tích chập và pooling để tạo ra các đặc trưng tầng cao.

Lấy đặc trưng từ mạng tích chập: Sau khi hình ảnh đi qua mạng tích chập, các đặc trưng tầng cao được lấy ra từ tầng cuối cùng của mạng tích chập. Điều này tạo ra một biểu diễn đặc trưng của hình ảnh với kích thước nhỏ hơn và chứa thông tin quan trọng về nội dung hình ảnh.

Chuẩn hóa và tái hình dạng đặc trưng: Đặc trưng được chuẩn hóa để đảm bảo cùng phạm vi giá trị hoặc cùng đơn vị chuẩn. Việc chuẩn hóa có thể bao gồm chuyển đổi giá trị về khoảng [0, 1] hoặc chuẩn hóa theo giá trị trung bình và độ lệch chuẩn. Đặc trưng cũng có thể được tái hình dạng để phù hợp với đầu vào của mô hình ngôn ngữ. Điều này có thể là vector 1D hoặc ma trận 2D phụ thuộc vào yêu cầu của mô hình ngôn ngữ.

4.4. Ứng Long Short-Term Memory để tạo chú thích hình ảnh

Xây dựng mô hình Long Short-Term Memory: Xác định số lớp Long Short-Term Memory, số đặc trưng đầu ra của Long Short-Term Memory, và số lượng

lớp ẩn (hidden layer) trong mô hình Long Short-Term Memory. Tạo một lớp nhúng (embedding layer) để biểu diễn từng từ dưới dạng vector. Xây dựng mô hình Long Short-Term Memory bằng cách kết hợp lớp nhúng với các lớp Long Short-Term Memory và các lớp mạng nơ-ron khác (nếu cần). Định nghĩa đầu vào của mô hình, bao gồm đặc trưng hình ảnh và câu chú thích đã được số hóa.

4.5. Huấn luyện mô hình

- Lựa chọn siêu tham số: nhóm chúng em sẽ chọn các siêu tham số sau:

Kích thước batch (Batch size): Đây là số lượng ảnh và các mô tả tương ứng được sử dụng trong mỗi lần cập nhật gradient. Lựa chọn batch size phù hợp có thể ảnh hưởng đến tốc độ huấn luyện và hiệu suất của mô hình. Thông thường, batch size càng lớn, quá trình huấn luyện càng nhanh, nhưng đòi hỏi bộ nhớ cũng như tài nguyên phần cứng lớn hơn.

Learning rate: Learning rate quyết định tốc độ học của mô hình trong quá trình cập nhật các trọng số. Một learning rate quá cao có thể dẫn đến việc bỏ qua điểm cực tiểu và mô hình không hội tụ, trong khi một learning rate quá thấp có thể làm chậm quá trình huấn luyện. Thường thì bạn có thể bắt đầu với một learning rate khá nhỏ (ví dụ: 0.001) và điều chỉnh nó dựa trên sự hội tụ của mô hình.

Số lượng epoch: Epoch là một vòng lặp qua toàn bộ tập dữ liệu huấn luyện. Số lượng epoch cần thiết để đạt được kết quả tốt có thể khác nhau cho từng mô hình và tập dữ liệu. Thông thường, quá nhiều epoch có thể dẫn đến overfitting (mô hình chỉ học thuộc lòng dữ liệu huấn luyện) trong khi quá ít epoch có thể không đủ để mô hình học các mẫu phức tạp trong dữ liệu.

Kích thước embedding: Kích thước embedding là số chiều của vector biểu diễn từ (word embedding). Kích thước embedding phải phù hợp với kích thước từ điển từ vựng của bạn. Một embedding quá nhỏ có thể làm mất mát thông tin.

4.6. Đánh giá mô hình

- Ta sẽ lựa chọn siêu tham số cho mô hình
    + model.optimizer.lr = 0.0001 (learning rate)
    + epochs=5
    + no_of_photos=5 (batch size)

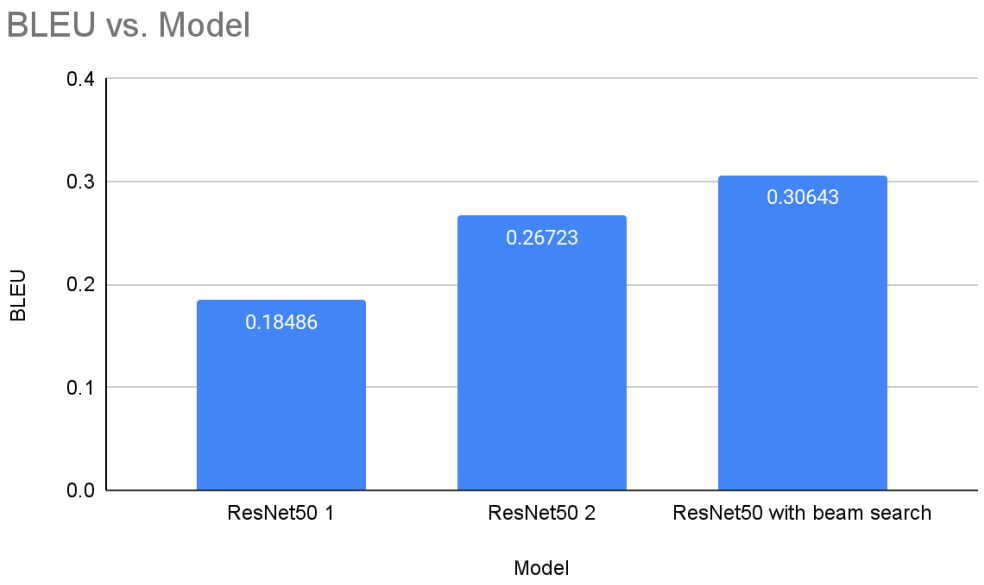
+ steps=len(train_encoded_captions)//no_of_photos (step for every epochs)

- Kết quả đánh giá

Model	Siêu tham số	BLEU	Sử dụng beam search với k = 3
ResNet50	learning rate = 0.0001, epochs = 3, batch size = 5 step = 1200	0.18486	Với bleu thấp nên ta sẽ không sử dụng beam search với siêu tham số này
ResNet50	learning rate = 0.0001, epochs = 5, batch size = 5 step = 1200	0.26723	0.30643

Bảng 1. Kết quả đánh giá của mô hình sử dụng ResNet50, Vgg16

4.7. Kết quả



Hình 15: Giá trị bleu tương ứng với từng mô hình

PHẦN KẾT LUẬN

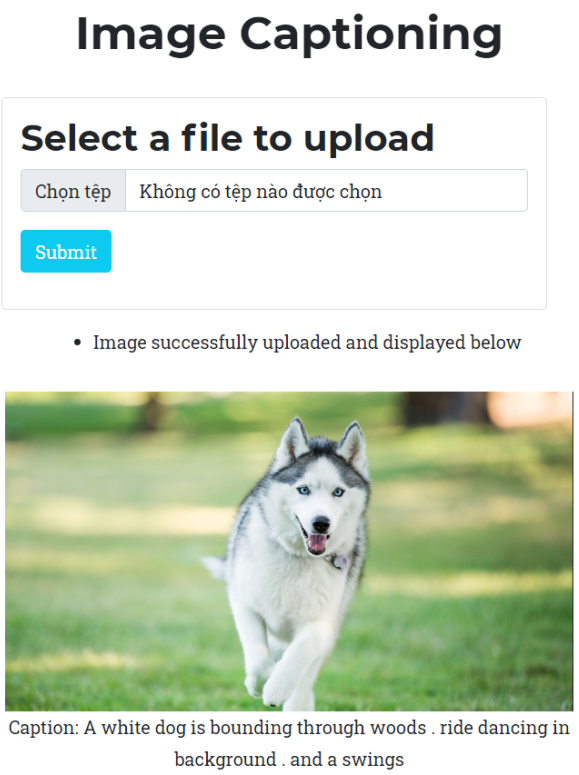
1. Tổng kết

1.1. Kiến thức tìm hiểu được

- Nắm được các kiến thức về học sâu, bao gồm kiến trúc mạng neural như Convolutional Neural Networks (tích chập) và Recurrent Neural Networks (RNN), thuật toán lan truyền ngược (backpropagation), và quá trình huấn luyện mô hình.
- Mạng hồi quy tích chập: Image Captioning kết hợp giữa xử lý hình ảnh và xử lý ngôn ngữ tự nhiên. Bạn cần hiểu về các phương pháp xử lý hình ảnh như trích xuất đặc trưng, phân loại, và nhận dạng đối tượng. Các kiến thức về xử lý hình ảnh, bao gồm Convolutional Neural Networks (tích chập), là cần thiết để trích xuất đặc trưng hình ảnh và tạo mô hình
- Xử lý ngôn ngữ tự nhiên: Image Captioning liên quan đến xử lý ngôn ngữ tự nhiên, vì vậy ta cần nắm vững các kiến thức về xử lý ngôn ngữ tự nhiên như xử lý văn bản, mô hình ngôn ngữ (language model), và mô hình seq2seq (sequence-to-sequence) như Long Short-Term Memory (Long Short-Term Memory).
- Mô hình kết hợp (hybrid models): Bài toán Image Captioning kết hợp cả xử lý hình ảnh và xử lý ngôn ngữ tự nhiên. Nên chúng ta cần hiểu cách kết hợp một mô hình học máy cho phần trích xuất đặc trưng hình ảnh và một mô hình ngôn ngữ cho phần sinh mô tả.
- Dữ liệu huấn luyện: Chúng ta cần phải hiểu được tập dữ liệu của bài toán như thế nào để đưa ra phương pháp xử lý phù hợp, ví dụ: một hình sẽ tương ứng có năm mô tả tương ứng.
- Phương pháp đánh giá: để đánh giá độ đo liên quan đến bài toán này ta cần có các kiến thức liên quan các độ đo như BLEU, nhằm tạo ra việc tạo ra mô ta chính xác đối với từng hình ảnh.

1.2. Chương trình đã làm được

- Sử dụng mô hình đã huấn luyện và đánh giá để tạo ra một ứng dụng web để thực hiện làm mô ta với từng bức hình mà ta muốn thử.



Hình 16: Kết quả thử nghiệm tạo chú thích hình ảnh trên ứng dụng web

2. Hạn chế của đề tài

- Do việc xử lý hình ảnh và việc tiền xử lý hình ảnh sẽ dẫn đến việc đưa ra dự đoán chưa chính xác.
- Tập dữ liệu với kích thước trung bình khoảng 8000 ảnh với 5 mô tả tương ứng với mỗi ảnh dẫn đến việc dự đoán chưa chính xác.
- Độ sâu của mô hình chưa tốt, có thể sử dụng độ sâu hơn để đảm bảo việc dự đoán tốt hơn.
- Hiểu và mô tả ngữ nghĩa chưa chính xác: Mô hình Image Captioning có thể gặp khó khăn trong việc hiểu và mô tả ngữ nghĩa chính xác của hình ảnh. Điều này có thể dẫn đến việc tạo ra các câu mô tả không đầy đủ hoặc không chính xác về nội dung hình ảnh.
- Tính phức tạp và tốn kém về thời gian tính toán: Mô hình Image Captioning thường có kiến trúc phức tạp và yêu cầu nhiều tài nguyên tính toán. Việc huấn luyện và dự đoán trên mô hình này có thể tốn nhiều thời gian và tài nguyên tính toán, đặc biệt khi xử lý ảnh có độ phân giải cao.

3. Hướng phát triển

- Sử dụng tập dữ liệu có kích thước lớn hơn tập dữ liệu như: Flickr30k, Coco,...
- Sử dụng mô hình có độ sâu phức tạp hơn nhằm mục đích tăng độ chính xác khi đưa ra giá trị dự đoán.
- Sử dụng Ensemble Learning: Kết hợp nhiều mô hình Image Captioning khác nhau để tạo ra mô hình Ensemble có khả năng tạo ra các mô tả hình ảnh đa dạng và chất lượng cao hơn. Ensemble Learning có thể giúp cải thiện độ chính xác và tính đa dạng của các mô hình Image Captioning.
- Tạo ra các bộ dữ liệu mới và phong phú hơn: Mở rộng tập dữ liệu Flickr8k hoặc thu thập các tập dữ liệu mới có độ phân loại chi tiết hơn, đa dạng hơn về nội dung hình ảnh và mô tả ngôn ngữ. Điều này giúp tăng tính tổng quát hóa và khả năng áp dụng của mô hình Image Captioning..

TÀI LIỆU THAM KHẢO

[1] Introduction to computer vision: History and applications, October 5, 2021, https://www.superannotate.com/blog/introduction-to-computer-vision.

[2] Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions, 18 August 2021, https://link.springer.com/article/10.1007/s42979-021-00815-1.

[3] Recurrent Neural Networks cheatsheet, By Afshine Amidi and Shervine Amidi, https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks#overview.

[4] Convolutional Neural Networks cheatsheet, By Afshine Amidi and Shervine Amidi ,https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks.

[5] Image Captioning – Chú thích dữ liệu hình ảnh bằng công nghệ học sâu, 13 Sept 2022, https://vinbigdata.com/kham-pha/image-captioning-chu-thich-du-lieu-hinh-anh-bang-cong-nghe-hoc-sau.html#Cac_kien_truc_cua_chu_thich_anh.

[6] CS231n: Deep Learning for Computer Vision, Stanford University, http://cs231n.stanford.edu/schedule.html.

[7] CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, Spring 2023, https://cs231n.github.io/.

[8] A Step by Step Backpropagation Example, by Matt Mazur, https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/.

PHỤ LỤC HÌNH ẢNH

[1], [2], [3] Mạng neural hồi quy cheatsheet: cheatsheet-recurrent-neural-networks.

[4], [5], [6] Mạng neural hồi quy cheatsheet: cheatsheet-recurrent-neural-networks.

[7] Mạng neural tích chập cheatsheet: cheatsheet-convolutional-neural-networks

[8] Mạng neural tích chập cheatsheet: cheatsheet-convolutional-neural-networks.

[9] Mạng neural tích chập cheatsheet: cheatsheet-convolutional-neural-networks.

[10] Mạng neural tích chập cheatsheet: cheatsheet-convolutional-neural-networks.

[11] Mạng neural tích chập cheatsheet: cheatsheet-convolutional-neural-networks.

[12] The Annotated ResNet-50: Resnet-50 Model architecture.

[13] A Brief Overview of Recurrent Neural Networks (RNN): Train RNN