TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO BÀI TẬP LỚN HỌC MÁY

Đề tài: Ứng dụng học sâu

cho bài toán tự động mô tả hình ảnh

Giảng viên hướng dẫn: **TS Thân Quang Khoát**

Giảng viên hướng dẫn: **TS Thân Quang Khoát**

Sinh viên thực hiện: **Đàm Minh Tiến** **20156599**

Lương Thành Long **20155970**

Phan Xuân Phúc **20156248**

Nguyễn Bình Minh **20156063**

Học kỳ: **20181**

MỤC LỤC

MỞ ĐẦU

Trong những năm gần đây, trí tuệ nhân tạo nói chung, học máy nói riêng đang trở thành một một xu hướng phát triển của công nghệ. Những bài toán lập trình có thể giải bằng cách cho máy tính học tập với dữ liệu có trước, từ đó giải quyết các lớp bài toán giống nhau mà không phải đi vào lập trình cụ thể.

Những năm gần đây, khi mà khả năng tính toán của các máy tính được nâng lên một tầm cao mới và lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn, học máy đã tiến thêm một bước dài và một lĩnh vực mới được ra đời gọi là học sâu (deep learning). Học sâu đã giúp máy tính thực thi những việc tưởng chừng như không thể vào 10 năm trước: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, hay thậm chí cả sáng tác văn hay âm nhạc.

Trong đó, tự tạo chú thích cho hình ảnh là một đề tài khó với việc yêu cầu kết hợp cả xử lí hình ảnh và xử lí ngôn ngữ ngôn ngữ tự nhiên. Tuy nhiên, đây cũng là một đề tài hay và có thể ứng dụng được nhiều vào thực tiễn như camera dẫn đường cho người mù hay hệ thống tự gán alt của facebook.

Chính vì những lý do trên nên chúng em chọn đề tài **ứng dụng học sâu cho bài toán tự động mô tả hình ảnh** cho bài tập lớn lần này.

Hà Nội, ngày 01 tháng 12 năm 2018

I: TỔNG QUAN

Bài toán tự động miêu tả hình ảnh là một bài toán hay và khó trong lĩnh vực học máy, yêu cầu hiểu biết về cả lĩnh vực xử lí ảnh lẫn xử lí ngôn ngữ tự nhiên. Bài toán đặt ra là đưa ra những câu miêu tả có ý nghĩa với bức ảnh được cho trước. Nếu một đứa trẻ quan sát phải hiểu và đủ vốn từ để có thể miêu tả được nội dung của bức ảnh thì không phải đứa trẻ nào cũng đưa ra những câu miêu tả giống nhau.

Trước đây, người ta có rất ít ý tưởng cho bài toán này , một số ý tưởng nổi trội như phát hiện các vật thể trong ảnh và cố gắng đi tìm mối tương quan giữa chúng, tuy nhiên kết quả chưa thực sự tốt, một phần khác do chưa có lượng dữ liệu đủ nhiều và tốt để huấn luyện. Với sự phát triển mạnh mẽ của phần cứng, đặc biệt là khả năng tính toán song song trên GPU và sự bùng nổ của internet dẫn tới dữ liệu lớn, các mạng nơ ron trở nên ưu việt trong các tác vụ xử lý ảnh. Năm 2012, Krizhevsky cùng các đồng nghiệp đưa ra mô hình mạng AlexNet, lần đầu thắng giải ILSVRC 2014 (một cuộc thi lớn trong tác vụ phân loại ảnh) đưa tỉ lệ phân loại lỗi top5-error giảm từ 26% xuống 16%, trở thành một cú hích lớn trong giới nghiên cứu. Cũng kể từ đó, mạng nơ-ron dần thay thế các phương pháp truyền thống trong các bài toán thị giác máy tính và cả xử lí ngôn ngữ. Bài báo đầu tiên ứng dụng học sâu vào tự động mô tả hình ảnh có lẽ phải kể đến "Show and tell: A neural image caption generator [1]" của O.Vinyals và các đồng nghiệp mà sau này trong các bài báo trích dẫn đến gọi là GoogleNIC. GoogleNIC với ý tưởng sử dụng một mạng nơ-ron tích chập (CNN) để trích xuất được đặc trưng của ảnh, kết hợp đi qua một một mạng hồi quy (RNN) để sinh câu miêu tả. Các phương pháp sau này cho bài toán mô tả hình ảnh đều dựa trên ý tưởng của phương pháp này.

Mô hình của nhóm chúng em đề xuất sử dụng mạng Resnet50 để trích xuất các đối tượng trong ảnh, đưa qua mạng LSTM để sinh từng từ, về cơ bản tương tự với mô hình của O.Vinyals et at. Trong báo cáo dưới đây, chúng em xin trình bày lí thuyết cơ bản về CNN và LSTM, mô hình sinh từ, cách đánh giá và hiệu chỉnh tham số, là kết quả thực hiện của nhóm trong bài tập lớn lần này.

Từ khóa: Image Captioning, Image Description, Explain Image, Convolutional Neural Networks, Recurrent Neural Networks, Long-Short Term Memory, Word Embeding, BLEU score.

Các công việc liên quan (Related works): Deep model for computer vision and natural language, Image-sentence retrieval, Generating novel sentence descriptions for images.

II: CƠ SỞ LÝ THUYẾT

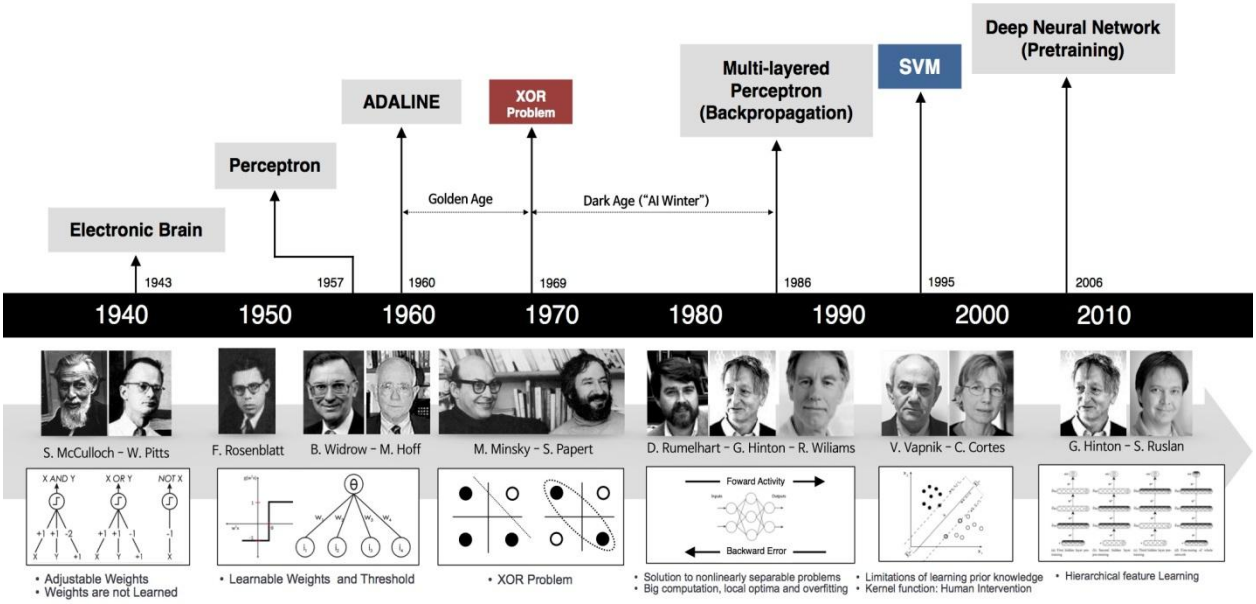
1. Học sâu (Deep learning)

Học sâu (hay deep learning) là một nhánh của học máy. Những năm gần đây, khi mà khả năng tính toán của các máy tính được nâng lên và lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn, học sâu nổi lên như một làn sóng công nghệ mới của thế giới. Học sâu với cấu trúc mạng nhiều lớp phi tuyến tính để trích xuất các đặc trưng quan trọng dựa trên việc huấn luyện một bộ dữ liệu mẫu đã phát triển rất nhanh và từng bước được nâng cao và gần tiệm cận với khả năng của con người trong một số lĩnh vực.

Nói đến học sâu và mạng noron không thể không nhắc tới các mô hình mạng tích chập (convolutional neural network). Chính nhờ sự thành công của mạng noron tích chập trong lĩnh vực phân loại ảnh đã giúp cho học sâu nổi lên và được biết đến nhiều hơn. Các mạng noron cơ bản đã được sử dụng vào những năm 1980. Tuy nhiên công nghệ này chỉ phát triển trong khoảng một chục năm trở lại đây với các thuật toán và cấu trúc mạng tối ưu hơn, sự hỗ trợ tính toán của các hệ thống máy tính cấu hình mạnh và hỗ trợ tính toán song song (GPU) cũng như số lượng dữ liệu được xử lý gán nhãn lớn (big data). Công nghệ học sâu này thực sự bùng nổ mạnh mẽ từ năm 2012 sau chiến thắng của Alex Krizhesky với mạng AlexNet nâng cấp từ mạng CNN cơ bản trong cuộc thi ILSVRC (ImageNet Large-Scale Visual Recognition Challenge – là một cuộc thi hằng năm trong lĩnh vực phân loại ảnh). Kết quả của AlexNet đã làm giảm tỉ lệ phân loại ảnh gán nhãn sai xuống từ 26% còn 16%, một kì tích tại thời điểm đó, tốt hơn nhiều so với các phương pháp học máy thời bấy giờ chủ yếu dựa trên SVM với trích chọn đặc trưng HoG, SIFT… Sau AlexNet, tất cả các mô hình giành giải cao trong các năm tiếp theo đều là các

deep networks (ZFNet 2013, GoogLeNet 2014, VGG 2014, ResNet 2015). Xu thế chung có thể thấy là các mô hình càng ngày càng sâu.

Những công ty công nghệ lớn cũng để ý tới việc phát triển các phòng nghiên cứu học sâu trong thời gian này. Rất nhiều các ứng dụng công nghệ đột phá đã được áp dụng vào cuộc sống hàng ngày. Google cho phép tìm kiếm hình ảnh, google dịch… Facebook sử dụng cho thuật toán tự động gán tag, chú thích… Amazon với hệ thống khuyến nghị sản phẩm. Cũng kể từ năm 2012, số lượng các bài báo khoa học về học sâu tăng lên theo hàm số mũ. Các blog về học sâu cũng tăng lên từng ngày.



Hình 1 Lược sử phát triển của học sâu

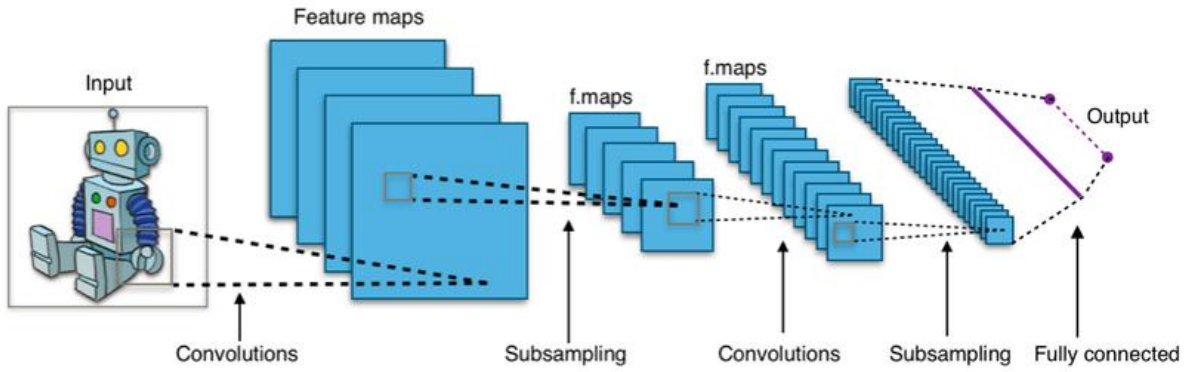
Rất nhiều những ý tưởng cơ bản của deep learning được đặt nền móng từ những năm 80-90 của thế kỷ trước, tuy nhiên deep learning chỉ đột phá trong khoảng 5-6 năm nay. Có nhiều nhân tố dẫn đến sự bùng nổ này:

- Sự ra đời ngày càng nhiều của các bộ dữ liệu lớn được gán nhãn và public.
- Khả năng tính toán song song tốc độ cao của GPU, TPU. Clould computing ngày càng phổ biến.
- Sự ra đời của các hàm kích hoạt liên quan làm hạn chế vấn đề vanishing gradient (ReLU…)
- Sự cải tiến của các kiến trúc: GoogLeNet, VGG, ResNet, … và các kỹ thuật transfer learning, fine tuning, enssemble.
- Nhiều kỹ thuật regularization mới: dropout, batch normalization, data augmentation.
- Nhiều thư viện mới hỗ trợ việc huấn luyện deep network với GPU: theano, caffe, mxnet, tensorflow, pytorch, keras, …
- Nhiều kỹ thuật tối ưu mới: Adagrad, RMSProp, Adam, …
- Cộng đồng chia sẻ, các cuộc thi và sự đầu tư của chính phủ cũng như các tập đoàn lớn.

***

2. Mạng nơ ron tích chập (Convolutional neural network)

Mạng nơ ron tích chập hay convolutional neural network (CNN) là mạng dạng tiếp thuận (feedforward) trong đó thông tin chỉ đi theo một chiều từ đầu vào đến đầu ra. CNN là một deep neural network (DNN). Hiểu đơn giản, nó cũng chính là một dạng artificial neural network (ANN), một multi-layer perceptron (MLP) nhưng mang thêm 1 vài cải tiến, đó là convolution và pooling.



Hình 2: Trong suốt quá trình huấn luyện, CNNs sẽ tự động học được các thông số cho các filter. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.

CNNs có tính bất biến và tính kết hợp cục bộ (Location Invariance and Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các gốc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể. Lớp tổng hợp (pooling layer) sẽ cho tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter.

2.1. Tầng tích chập (Convolutional layers)

Tầng tích chập được dùng để phát hiện và trích xuất đặc trưng - chi tiết của ảnh. Tầng đầu tiên trong một mạng nơ ron tích chập hẳn là một tầng tích chập. Đây chính là cải tiến đáng kể của mạng LeNet (mạng CNN đầu tiên) so với các mạng MLP truyền thống.

Giống như các lớp ẩn khác, lớp tích chập lấy dữ liệu đầu vào, thực hiện các phép chuyển đổi để tạo ra dữ liệu đầu vào cho lớp kế tiếp (đầu ra của lớp này là đầu vào của lớp sau). Phép biến đổi được sử dụng là phép tính tích chập. Mỗi lớp tích chập

chứa một hoặc nhiều bộ lọc - bộ phát hiện đặc trưng (filter - feature detector) cho phép phát hiện và trích xuất những đặc trưng khác nhau của ảnh. Đặc trưng của ảnh là gì? Đặc trưng ảnh là những chi tiết xuất hiện trong ảnh, từ đơn giản như cạnh, hình khối, chữ viết tới phức tạp như mắt, mặt, chó, mèo, bàn, ghế, xe, đèn giao thông, v.v.. Bộ lọc phát hiện đặc trưng là bộ lọc giúp phát hiện và trích xuất các đặc trừng của ảnh, có thể là bộ lọc góc, cạnh, đường chéo, hình tròn, hình vuông, v.v.

Bộ lọc ở lớp tích chập càng sâu thì phát hiện các đặc trừng càng phức tạp. Độ phức tạp của đặc trưng được phát hiện bởi bộ lọc tỉ lệ thuận với độ sâu của lớp tích chập mà nó thuộc về. Trong mạng CNN, những lớp tích chập đầu tiên sử dụng bộ lọc hình học (geometric filters) để phát hiện những đặc trưng đơn giản như cạnh ngang, dọc, chéo của bức ảnh. Những lớp tích chập sau đó được dùng để phát hiện đối tượng nhỏ, bán hoàn chỉnh như mắt, mũi, tóc, v.v. Những lớp tích chập sâu nhất dùng để phát hiện đối tượng hoàn hỉnh như: chó, mèo, chim, ô tô, đèn giao thông, v.v.

Nếu ta có tập dữ liệu huấn luyện lớn và hiệu năng tính toán cao, với những tập ảnh kích thước lớn và nhiều chi tiết phức tạp, các bộ lọc cạnh có thể được huấn luyện tự động từ tập dữ liệu. Nghĩa là các giá trị của ma trận lọc được coi như tham số của một mạng nơ-ron và huấn luyện (sử dụng back-propagation chẳng hạn) để có một tập giá trị tối ưu. Với cách tiếp cận này, các bộ lọc tạo ra có thể phát hiện không chỉ cạnh đứng hay ngang mà còn có thể những cạnh nghiêng một góc lẻ như 40^o, 45^o hoặc 70^o.

Stride and Padding

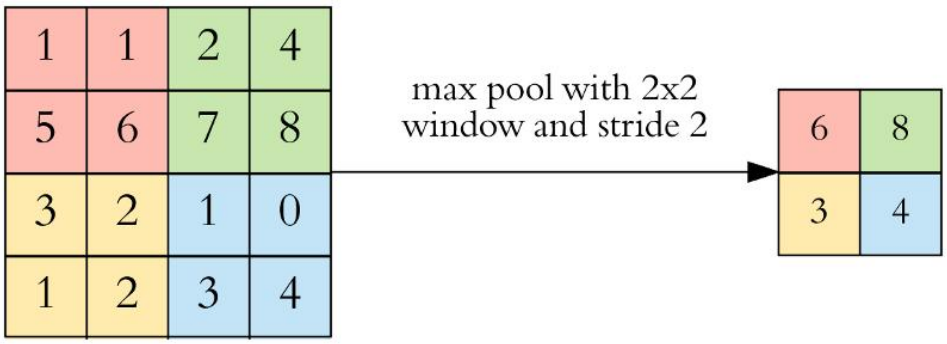
Ảnh hưởng của phép tích chập:

- Nếu ta nhân chập ma trận đầu vào kích thước $n \times n \times n$ với bộ lọc kích thước $f \times f \times f$, ta thu được kết quả là một ma trận kích thước $(n-f+1) \times (n-f+1)$. Mỗi một lần áp dụng phép nhân chập, kích thước của ảnh bị giảm xuống, và vì thế chúng ta chỉ có thể thực hiện nó một vài lần trước khi ảnh trở nên quá nhỏ.
- Điểm ảnh ở khoảng trung tâm của ma trận đầu vào được nhân rất nhiều lần tích chập, trong khi các điểm ảnh bên ngoài hoặc biên chỉ được nhân 1-2 lần. Vì thế chúng ta đánh mất rất nhiều thông tin (có thể quan trọng) tại các vùng gần cạnh của ảnh.

Để khắc phục hai nhược điểm trên, một đường viền phụ (padding) được thêm vào xung quanh ma trận đầu. Việc thêm đường viền phụ làm tăng kích thước của ma trận đầu vào, dẫn tới tăng kích thước ma trận đầu ra. Từ đó độ chênh lệch giữa ma trận đầu ra với ma trận đầu vào gốc giảm. Những ô nằm trên cạnh/ góc của ma trận đầu vào gốc cũng lùi sâu vào bên trong hơn, dẫn tới được sử dụng nhiều hơn trong việc tính toán ma trận đầu ra, tránh được việc mất mát thông tin.

2.2. Tầng tổng hợp (Pooling layers)

Mục đích của pooling rất đơn giản: làm giảm số siêu tham số (hyperparameter) cần phải tính toán, từ đó giảm thời gian tính toán, tránh hiện tượng quá khớp (overfitting). Loại pooling thường gặp nhất là max pooling, lấy giá trị lớn nhất trong một cửa sổ pooling. Pooling hoạt động gần giống với convolution, cũng có 1 cửa sổ trượt gọi là pooling window, cửa sổ này trượt qua từng giá trị của ma trận dữ liệu đầu vào, thường là các đặc trưng (feature map) trong lớp tích chập, chọn ra một giá trị từ các gía trị nằm trong cửa sổ trượt (với max pooling ta sẽ lấy giá trị lớn nhất).



*Hinh 3: Max pooling window có kích thước là 2 * 2, stride = 2*

2.3. Tầng kết nối đầy đủ (Fully connected layers)

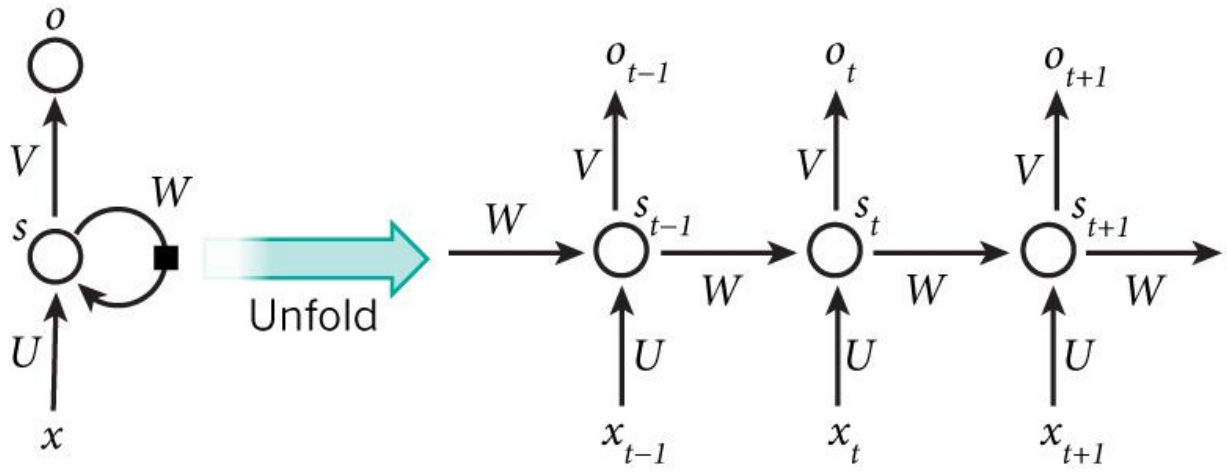
Layer này cũng chính là một fully connected ANN. Thường thì sau các lớp Conv + Pooling thì sẽ là 2 lớp fully connected, 1 layer để tập hợp các feature layer mà ta đã tìm ra, chuyển đổi dữ liệu từ 3D, hoặc 2D thành 1D, tức chỉ còn là một vector. Còn một layer nữa là output, số neuron của layer này phụ thuộc vào số output mà ta muốn tìm ra. Giả sử với tập dữ liêu MNIST chẳng hạn, ta có tập các số viết tay từ 0 -> 9. Vậy output sẽ có số neuron là 10.

***

3. Mạng hồi quy (Recurrent neural network)

Nhìn lại cấu trúc của mạng nơ-ron nhân tạo, ta thấy cấu tạo luôn gồm ba tầng: tầng đầu vào (input layer), tầng ẩn (hidden layer) và tầng đầu ra (output). Vấn đề là tầng đầu vào và tầng đầu ra này là độc lập. Giả sử với bài toán sinh chuỗi mô tả, đầu ra của mạng lại tại thời điểm này lại tiếp tục là đầu vào của mạng tại thời điểm tiếp theo. Chính vì vậy nên cấu trúc mạng truyền thống không phù hợp với những bài toán dạng chuỗi. Và như vậy, mạng hồi quy – recurrent neural network (RNN) ra đời với ý tưởng chính là sử dụng bộ nhớ để lưu lại thông tin từ từ những

bước tính toán xử lý trước để dựa vào đó có thể đưa ra dự đoán chính xác nhất cho bước tính toán hiện tại. Về cơ bản, một RNN có dạng như sau:



Hình 4: Cấu trúc một RNN cơ bản

Nhìn vào hình, ta có thể miêu tả quá trình tính toán bên trong một RNN, ta xét tại trạng thái t:

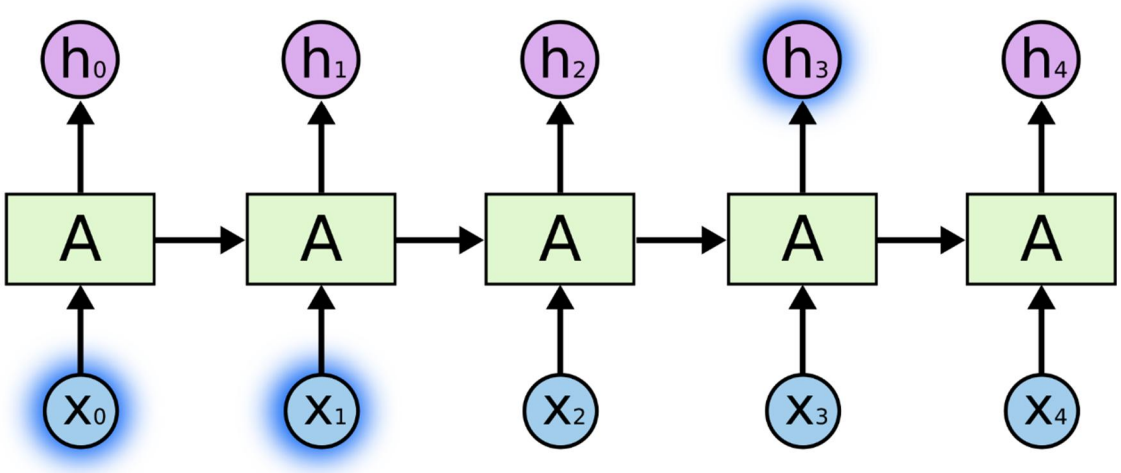
- x_t là đầu vào tại bước thứ t.
- s_t là trạng thái ẩn tại bước thứ t. Nó chính là bộ nhớ của mạng. s_t được tính toán thông qua cả đầu vào tại chính bước đó t và trạng thái ẩn trước đó t-1. $s_t = f(Ux_t + Ws_{t-1})$ trong đó, hàm f có thể là hàm tanh hoặc ReLU. Để có thể tính toán được cho bước đầu tiên, thường khởi tạo s = 0.
- o_t là đầu ra tại bước thứ t. Ví dụ, ta muốn dự đoán từ tiếp theo có thể xuất hiện trong câu thì o_t chính là một véc tơ xác suất các từ trong danh sách từ vựng $o_t = softmax(Vs_t)$.

4. Mạng bộ nhớ dài hạn – ngắn hạn (Long short term memory)

4.1. Vấn đề của mất mát đạo hàm

Một điểm nổi bật của RNN chính là ý tưởng kết nối các thông tin phía trước để dự đoán cho hiện tại. Việc này tương tự như ta sử dụng các cảnh trước của bộ phim để hiểu được cảnh hiện thời. Nếu mà RNN có thể làm được việc đó thì chúng sẽ cực kì hữu dụng, tuy nhiên liệu chúng có thể làm được không? Câu trả lời là còn tùy.

Đôi lúc ta chỉ cần xem lại thông tin vừa có thôi là đủ để biết được tình huống hiện tại. Ví dụ, ta có câu: "các đám may trên bầu trời" thì ta chỉ cần đọc tới "các đám may trên bầu" là đủ biết được chữ tiếp theo là "trời" rồi. Trong tình huống này, khoảng cách tới thông tin có được cần để dự đoán là nhỏ, nên RNN hoàn toàn có thể học được.



Hình 5: Quá trình học tại một RNN

Nhưng trong nhiều tình huống ta buộc phải sử dụng nhiều ngữ cảnh hơn để suy luận. Ví dụ, dự đoán chữ cuối cùng trong đoạn: "I grew up in France… I speak

fluent French.”. Rõ ràng là các thông tin gần (”I speak fluent”) chỉ có phép ta biết được đằng sau nó sẽ là tên của một ngôn ngữ nào đó, còn không thể nào biết được đó là tiếng gì. Muốn biết là tiếng gì, thì ta cần phải có thêm ngữ cảnh “I grew up in France” nữa mới có thể suy luận được. Rõ ràng là khoảng cách thông tin lúc này có thể đã khá xa rồi.

Thật không may là với khoảng cách càng lớn dần thì RNN bắt đầu không thể nhớ và học được nữa. Về mặt lý thuyết, rõ ràng là RNN có khả năng xử lý các phụ thuộc xa (long-term dependencies). Chúng ta có thể xem xét và cài đặt các tham số sao cho khéo là có thể giải quyết được vấn đề này. Tuy nhiên, đáng tiếc trong thực tế RNN có vẻ không thể học được các tham số đó. Vấn đề này đã được khám phá khá sâu bởi Hochreiter (1991) [tiếng Đức] và Bengio, et al. (1994), trong các bài báo của mình, họ đã tìm được nhưng lý do căn bản để giải thích tại sao RNN không thể học được.

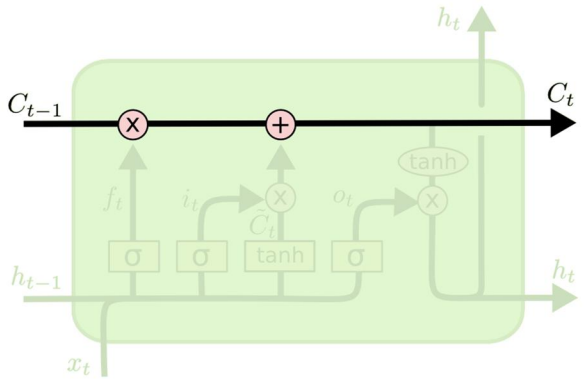
4.2. Những cải tiến mới của LSTM

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter & Schmidhuber (1997), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào. Mọi mạng hồi quy đều có

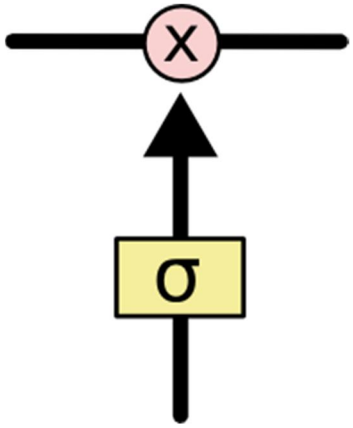
dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng tanh.

Chìa khóa của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ. Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.



Hình 6: Một cell LSTM

LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate).



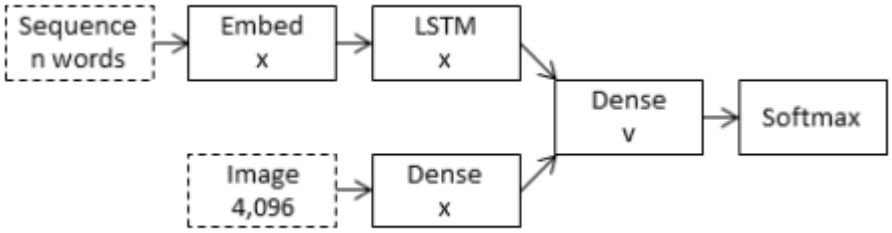
Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân. Tầng sigmoid sẽ cho đầu ra là một số trong khoảng $[0, 1]$, mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 0 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 1 thì có nghĩa là cho tất cả các thông tin đi qua nó. Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

III: BÀI TOÁN MÔ TẢ HÌNH ẢNH

1. Ý tưởng chính

Dựa trên các mô hình đã công bố, ta đưa ra một vài ý tưởng đặc trưng xuyên suốt: 1) Kết hợp mạng tích chập CNN và mạng hồi quy LSTM, 2) CNN trích xuất dữ liệu ảnh (encoder), 3) LSTM sinh câu miêu tả (encoder) 4) Decoder thông qua một tầng fully connected, end-to-end model: image -> sentence và 5) Maxmize P(S|I) - với P(S|I,w) là xác suất của câu miêu tả S nếu biết ảnh I, và tham số w.

Như vậy, ta cần đi tìm bộ tham số w^* sao cho $w^* = argmax \sum_{(I,S)} \log p\,(S|I,w)$



Mô hình sử dụng theo kiểu "merge model", tức các véc-tơ ảnh và text sau khi được xử lí và trích xuất thông qua các tầng LSTM thành hai véc-tơ thì sẽ được đưa qua tầng Merge Layer để "trộn lại" và cuối cùng sử dụng dense layer là một tầng kết nối với hàm kích hoạt softmax để tính toán ra output là từ tiếp theo trong câu. Trong mô hình này, tại các thời điểm, tầng LSTM và tầng trích xuất đặc trưng ảnh là hoàn toàn độc lập, LSTM không nhận bất cứ véc-tơ input nào của ảnh và chỉ xử lí dữ liệu văn bản, sự kết hợp chỉ xảy ra sau khi trộn hai véc tơ này lại tại layer merge.

2. Dữ liệu và tiền xử lý dữ liệu

Dữ liệu được nhóm sử dụng là bộ Flickr 8k, gồm 8092 ảnh với kích thước khác nhau và gần 40460 câu miêu tả, ~5 câu miêu tả cho một hình ảnh. Chi tiết mô tả đầy đủ về bộ dữ liệu được trích trong bài báo: "Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics" năm 2013. Trích dẫn lời nhóm tác giả: "*We introduce a new benchmark collection for sentence-based image description and search, consisting of 8,000 images that are each paired with five different captions which provide clear descriptions of the salient entities and events … The images were chosen from six different Flickr groups, and tend not to contain any well-known people or locations, but were manually selected to depict a variety of scenes and situations*". Flickr 8k là bộ dữ liệu tốt và phù hợp với lượng tài nguyên tính toán của nhóm.

Bộ dữ liệu flickr 8k được chia theo tỉ lệ 6:1:1 cho train:validate:test. Bộ train gồm 6000 ảnh và 30000 câu miêu tả, bộ validate và test tương tự nhau gồm 1000 ảnh và 5000 câu miêu tả.

Tiền xử lý dữ liệu ảnh: Dữ liệu ảnh trước khi đưa vào mô hình huấn luyện cần được trích xuất đặc trưng đưa về dạng véc-tơ. Với 8000 ảnh ban đầu có kích thước tùy ý -> resize về kích thước 224x224 -> đưa qua mạng resnet50 pre-train trên tập imagenet (bỏ layer fully connected cuối) -> véc-tơ 4096 thành phần.

Tiền xử lý dữ liệu văn bản: Dữ liệu văn bản của bộ flickr nằm trong 4 file text:

- Flickr8k.token.txt – chứa các miêu tả của ảnh. Cột đầu tiên là id của ảnh và số thứ tự (0->4), cột thứ hai là câu miêu tả.
- Flickr_8k.trainImages.txt – chứa id các ảnh dùng để train.

- Flickr_8k.devImages.txt – chứa id các ảnh dùng để validate.
- Flickr_8k.testImages.txt – chứa id các ảnh dùng để test.

Tại bước tiền xử lí ta chỉ quan tâm đến file Flickr8k.token.txt. Các câu miêu tả được gán với id của ảnh nằm trong một file text, ta đọc file này, trích chọn các câu miêu tả và làm sạch:

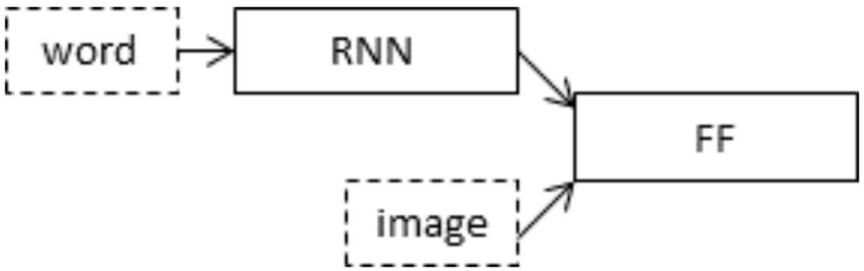
- Chuyển tất cả các từ thành chữ viết thường
- Loại bỏ các dấu câu
- Loại bỏ các stop word, từ ngắn (độ dài từ 1 trở xuống)
- Loại bỏ các từ có chứa số

3. Xây dựng mô hình và hiệu chỉnh

Sau khi tham khảo và nhiều lần thử nghiệm, nhóm đề xuất một mô hình học sâu tạm gọi "merge-model" dựa trên bài báo của Marc Tanti, et al. năm 2017:

- Where to put the Image in an Image Caption Generator, 2017.
- What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator, 2017.

Trong bài báo, tác giả đưa ra một lược đồ khái quát:



Chúng ta mô tả mô hình này thành 3 phần:

- Photo Feature Extractor: đây là chính là mô hình mạng CNN dùng để trích xuất ảnh, đã được pre-train trên bộ dữ liệu lớn và bỏ đi layer FC cuối để phân loại. Nhiệm vụ của phần này là trích xuất những đặc trưng của ảnh.
- Sequence Processor: đây là tầng word emdedding để xử lý đầu vào text, chính là một tầng LSTM.
- Decoder: cả hai tầng Photo Feature Extractor và Sequence Processor đều output ra một véc tơ có độ dài cố định, tầng dense sẽ merge chúng lại và đưa ra predicted cuối cùng.

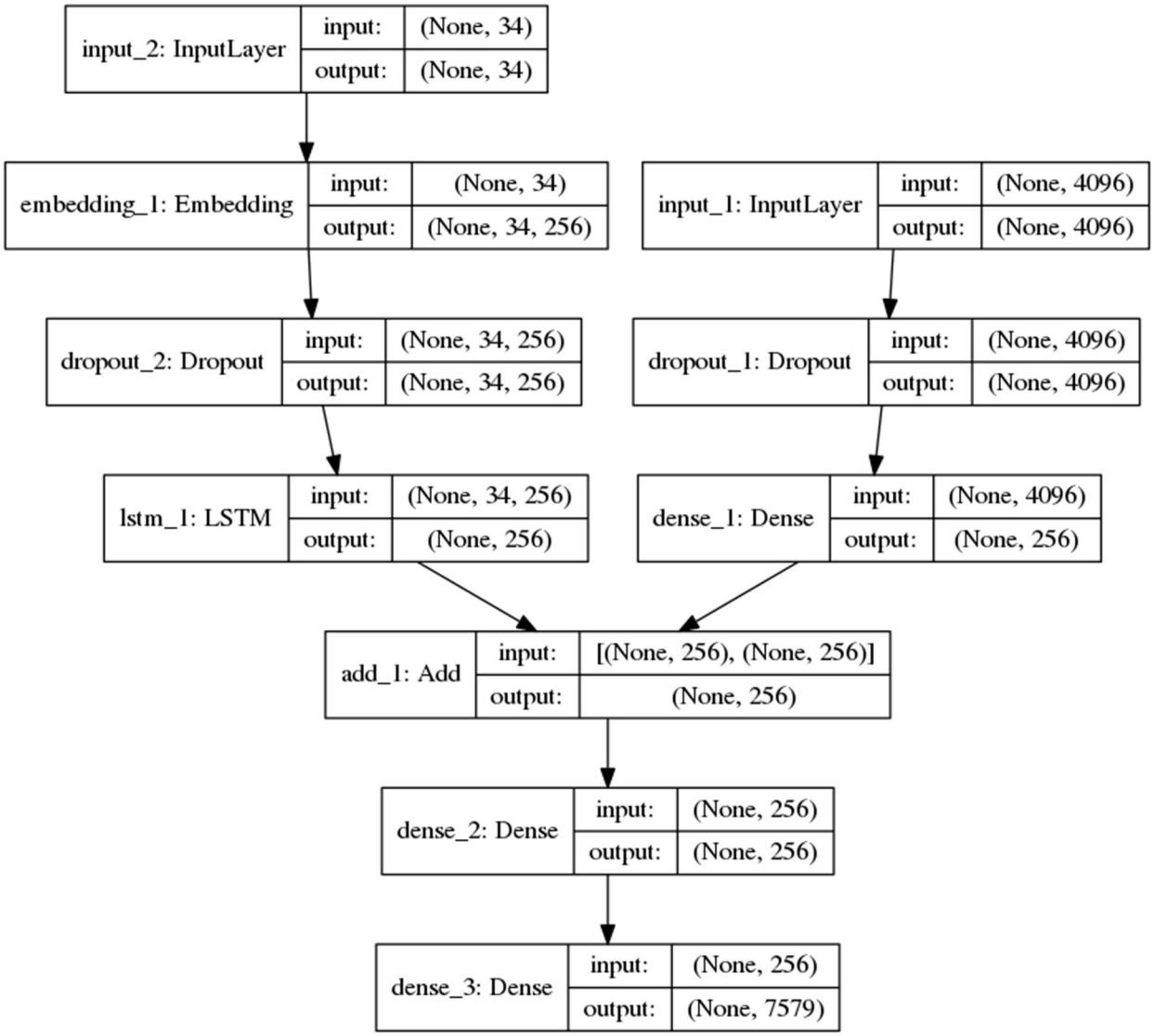
Cụ thể hóa, ta xây dựng mô hình mô tả hình ảnh trên nền keras như dưới hình vẽ:

Tầng Photo Feature Extractor nhận đầu vào là một véc-tơ 4096 thành phần, được xử lý qua một lớp fully connected và đưa về một véc-tơ 256 thành phần.

Tầng Sequence Processor nhận đầu vào là các câu với độ dài cố định (34 từ), sau khi đưa qua tầng embedding, rồi đưa qua tầng LSTM với 256 đơn vị nhớ (memory units) để đưa ra một véc-tơ 256 thành phần.

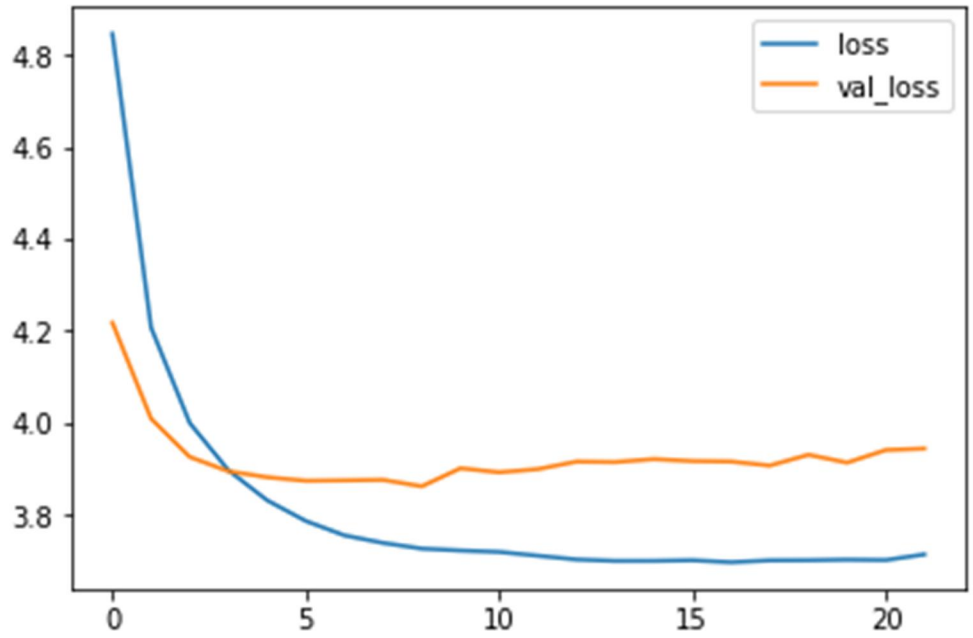
Các tầng Dropout được thêm vào giúp mô hình có thể tránh được overfitting và quá trình huấn luyện nhanh hơn.

Cuối cùng, các véc-tơ output 256 thành phần của Photo Feature Extractor và Sequence Processor được merge vào một véc tơ (sử dụng phép cộng véc tơ). Véc-tơ tổng đi qua dense layer cuối với activate function là softmax tính xác suất của từ tiếp theo.



Mô hình sử dụng hàm mất mát là categorical_crossentropy, optimizer Adam.

Sau khi train thử nghiệm trên 24 epochs, trên cloud của kaggle mất ~5 tiếng, biểu đồ loss theo epochs của mô hình như sau:



Ta có thể thấy loss của tập train tiệm cận tới giá trị khoảng 3.6. Validate loss thấp nhất khoảng 3.7 ở epochs thứ 8, sau đó tăng dần có dấu hiệu của overfitting.

4. Đánh giá mô hình và kết quả

Phương pháp đánh giá: Có nhiều phương pháp đánh giá một mô hình sinh ngôn ngữ như ROUGE, BLEU… Trong bài tập lớn lần này, ta chọn BLEU vì:

- ✓ Nhanh, không đắt đỏ, không phụ thuộc ngôn ngữ, tính tương quan cao với sự đánh giá của con người, tốn ít tài nguyên
- ✓ Tiêu chí đánh giá phù hợp và dễ hiểu: câu mà máy tính sinh ra càng giống với câu con người đặt ra thì càng dễ hiểu

- ✓ Đánh giá những khía cạnh: tính đầy đủ, tính trung thực và tính trôi chảy của câu máy sinh ra so với 1 tập câu tham khảo của con người.

Cách tính:

Score Calculation in BLEU

$$\text{Unigram precision } \mathbf{P} = \frac{m}{w_t}$$

$$\text{Brevity penalty } p = \begin{cases} 1 & \text{if } c > r \\ e^{\left(1-\frac{r}{c}\right)} & \text{if } c \leq r \end{cases}$$

$$\text{BLEU} = p.\, e^{\sum_{n=1}^{N} \left(\frac{1}{N} * \log Pn\right)}$$

Trong đó:

- m: min (số lần cụm n-gram xuất hiện trong candidate, max(số lần cụm n-gram xuất hiện trong 1 reference))
- w_t: số lượng các cụm n-gram xuất hiện trong candidate

Kết quả: Mô hình cho điểm BLEU score như sau:

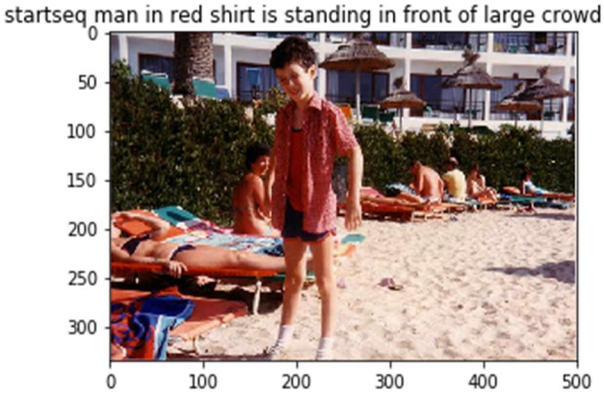
BLEU-1: 0.542805

BLEU-2: 0.301714

BLEU-3: 0.207351

BLEU-4: 0.095704

Các trường hợp đúng:



Các trường hợp sai:

IV: KẾT LUẬN

1. Kết quả đạt được

- Xây dựng và huấn luyện mạng nơ ron giải quyết được lớp bài toán mô tả hình ảnh
- Tìm hiểu các cách giải quyết khác trong các bài báo khoa học quốc tế
- Nắm được cách đánh giá và cải thiện các mô hình học máy
- Nâng cao các kiến thức về học máy, mạng nơ ron trong xử lí ảnh và ngôn ngữ tự nhiên
- Hiểu biết thêm các kĩ thuật tối ưu và các hàm mất mát thường dùng trong thực tế
- Kinh nghiệm khi lập trình với các thư viện học máy trong python, tiền xử lí dữ liệu, visuallize, viết báo cáo…
- Phong cách làm việc nhóm trong nghiên cứu
- Source code dự án: https://github.com/damminhtien/Deep-learning-Image-Caption-Generator

2. Phân chia công việc

Đàm Minh Tiến	✓ Lên ý tưởng và phân công công việc ✓ Lập trình ✓ Thử nghiệm và hiệu chỉnh các tham số ✓ Viết báo cáo, thuyết trình
Lương Thành Long Phan Xuân Phúc	✓ Đề xuất các chiến lược huấn luyện ✓ Xây dựng mô hình

Nguyễn Bình Minh	✓ Tiền xử lý dữ liệu ✓ Đề xuất các metric đánh giá mô hình

3. Hướng phát triển

- Tiếp tục tìm hiểu và nâng cao kiến thức về học máy
- Cải thiện kết quả của mô hình cũ
- Tìm hiểu và áp dụng các cách làm mới
- Huấn luyện và đánh giá với lượng dữ liệu lớn hơn như bộ Flickr 30k, MSCOCO…
- Xây dựng thành sản phẩm ứng dụng vào thực tiễn

3. Lời kết

Bài toán tự động mô tả hình ảnh là một bài toán hay và khó đòi hỏi kiến thức về cả xử lí ảnh và xử lí ngôn ngữ. Với sự phát triển của những mô hình học sâu trong những năm gần đây, các phương pháp với mạng nơ ron nhân tạo đang cho kết quả tốt và dẫn đầu trong tác vụ. Mô hình mạng nơ ron học sâu với sự kết hợp của mạng CNN và mạng LSTM được huấn luyện trên bộ dữ liệu flickr 8k với 8000 ảnh và ~40000 câu miêu tả đã cho kết quả tốt, giải quyết được bài toán đặt ra.

Chúng em xin gửi lời cảm ơn chân thành tới thầy **Thân Quang Khoát**, người đã hướng dẫn chúng em những kiến thức để có thể hoàn thành bài tập lớn này. Chúng em sẽ cố gắng tiếp tục học tập và nghiên cứu để đạt được những kết quả tốt hơn trong tương lai.

Tài liệu tham khảo

[1] Marc Tanti, Albert Gatt. Where to put the Image in an Image Caption Generator. arXiv preprint arXiv:1703.09137, 2018.

[2] Marc Tanti, Albert Gatt, Kenneth P. Camilleri. What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator? arXiv preprint arXiv:1708.02043, 2017

[1] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. arXiv preprint arXiv:1502.03044, 2016.

[2] Andrej Karpathy, Li Fei-Fei Deep Visual-Semantic Alignments for Generating Image Descriptions. arXiv preprint arXiv:1412.2306, 2015.

[3] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. arXiv preprint arXiv:1411.4555, 2014.

[4] Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, Serge Belongie. Learning to Evaluate Image Captioning.

[5] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Alan L. Yuille. Explain Images with Multimodal Recurrent Neural Networks. arXiv preprint arXiv:1410.1090, 2014.

[6] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, Trevor Darrell. Long-term

Recurrent Convolutional Networks for Visual Recognition and Description. arXiv preprint arXiv:1411.4389, 2016.

[7] Xinlei Chen, C. Lawrence Zitnick. Learning a Recurrent Visual Representation for Image Caption Generation. arXiv preprint arXiv:1411.5654, 2016.