BÁO CÁO CÔNG VIỆC TUẦN

Đề tài: Image captioning

Người thực hiện: Nguyễn Minh Đức

Hà Nội, tháng 6 năm 2021

Mục lục

Lời nói đầu

Web cung cấp một lượng lớn thông tin, bao gồm rất nhiều văn bản, nhưng nó ngày càng bị chi phối bởi thông tin trực quan, cả tĩnh (hình ảnh) và động (video). Tuy nhiên, nhiều thông tin thị giác đó không thể truy cập được đối với những người khiếm thị hoặc chậm tốc độ ternet cấm tải hình ảnh. Chú thích hình ảnh, được nội dung thêm theo cách thủ công nhà cung cấp (thường bằng cách sử dụng thẻ Alt-text HTML), là một cách để làm cho nội dung này nhiều hơn có thể truy cập được, để hệ thống chuyển văn bản thành giọng nói có thể được áp dụng để tạo ra ngôn ngữ tự nhiên tập lệnh của hình ảnh và video. Tuy nhiên, các mô tả hình ảnh do con người quản lý hiện có được thêm vào chỉ cho một phần rất nhỏ hình ảnh web. Vì vậy, có sự quan tâm lớn đến việc phát triển các phương pháp để tự động tạo mô tả hình ảnh.

Mô tả hình ảnh tự động, còn được cộng đồng nghiên cứu gọi là chú thích hình ảnh, có thể được định nghĩa là nhiệm vụ tự động tạo mô tả về hình ảnh bằng cách sử dụng tự nhiên ngôn ngữ. Đó là một vấn đề rất thách thức bao gồm hai loại vấn đề: vấn đề hiểu một hình ảnh, đó là nhiệm vụ Thị giác máy tính (CV) và vấn đề tạo ra một mô tả có ý nghĩa và đúng ngữ pháp về hình ảnh, đây là một loại Nhiệm vụ Xử lý Ngôn ngữ Tự nhiên (NLP). Do đó, để giải quyết công việc này, cần để thúc đẩy nghiên cứu trong hai lĩnh vực, CV và NLP, cũng như thúc đẩy hợp tác của cả hai cộng đồng để giải quyết các vấn đề cụ thể phát sinh khi kết hợp cả hai nhiệm vụ.

Trong đề tài bài tập lớn này, em xin được trình bày về hướng giải quyết bài toán này thông qua những gì em đã tìm hiểu được. Nội dung bao gồm các chương:

Chương 1. Giới thiệu về bài toán Image Captioning.

Chương 2.Hướng tiếp cận bài toán.

Chương 3.Tập dataset.

Chương 4.Chi tiết mô hình giải quyết bài toán.

Chương 1. Giới thiệu bài toán Image Captioning

1. Định nghĩa bài toán?

```
<start> two brown bears lying together and relaxing on a rock <end>
```

Hình 1 Hình ảnh chú thich

Image Captioning (còn được gọi là **gắn thẻ hình ảnh tự động** hoặc **lập chỉ mục ngôn ngữ**) là quá trình hệ thống máy tính tự động gán siêu dữ liệu dưới dạng phụ đề hoặc từ khóa cho hình ảnh kỹ thuật số

2. Ứng dụng

Các ứng dụng của Image Captioning cho các công việc trong cuộc sống:

- Xe tự lái (self driving cars): chắc hẳn các bạn chả xa lạ gì với từ khóa trên phải không nào, lái xe tự động là một bài toán lớn với rất nhiều những thử thách mà được các ông lớn về công nghệ luôn tìm hướng cải tiến, giải quyết các bài toán trong đó và nếu chúng ta chú thích được đúng cảnh xung quanh ô tô thì thật tuyệt với, nó sẽ thúc đẩy hệ thống sẽ tự lái với những thông tin được cung cấp rất có giả trị.
- Hỗ trợ người mù (aid to the blind): chúng ta có thể tạo ra một sản phẩm hỗ trợ người mù hướng dẫn họ khi họ di chuyển qua đường, đi lại mà không cần sự hỗ trợ của người khác. Trước tiên chúng ta có thể chuyển hình ảnh xung quanh họ thành đoạn

text sau đó từ text sẽ chuyển thành lời nói để họ có thể chủ động trong việc xử lý các tình huống với những thông tin được cung cấp.

- Camera giám sát: ngày nay trên các hệ thống đường đi lại, tòa nhà công.. của các quốc gia đã triển khai các hệ thống camera giám sát, nếu chúng ta cũng tạo được những phụ đề phù hợp thì thì có thể báo động ngay những hành động không tốt đang diễn ra như cảnh báo nguy cơ tai nạn, trộm cướp, cháy nổ...

Chương 2. Hướng tiếp cận bài toán

Input là ảnh và **output** là text, ví dụ "man in black shirt is playing guitar".

Nhìn chung các mô hình machine learning hay deep learning đều không xử lý trực tiếp với text như 'man', 'in', 'black',… mà thường phải quy đổi (encode) về dạng số. Từng từ sẽ được encode sang dạng vector với độ dài số định, phương pháp đấy gọi là word embedding. Nhìn thấy output là text nghĩ ngay đến RNN và sử dụng mô hình **LSTM/GRU**.

Input là ảnh thường được extract feature qua pre-trained model với dataset lớn như ImageNet và model phổ biến như VGG16, ResNet, quá trình được gọi là embedding và output là 1 vector.

1. Ý tưởng

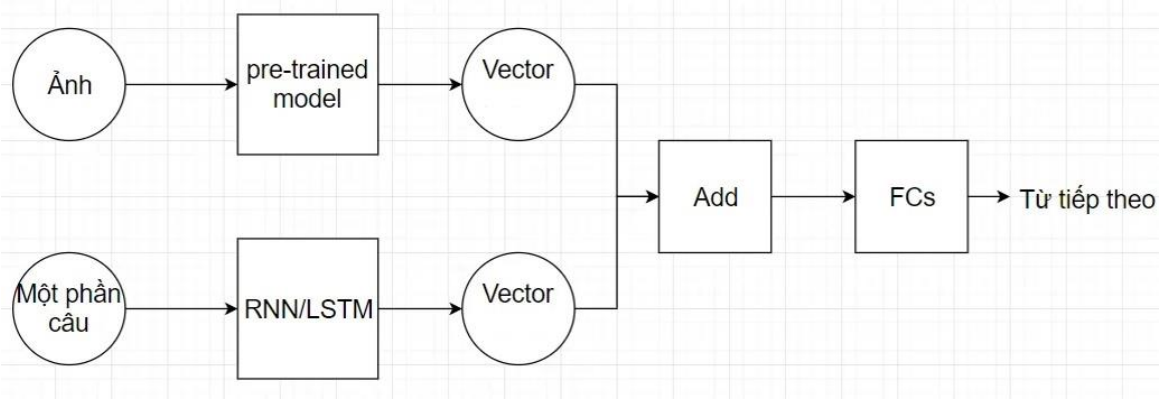
Ý tưởng sẽ là dùng embedding của ảnh và dùng các từ phía trước để dự đoán từ tiếp theo trong caption.

Để có thể dự đoán được từ tiếp theo, em xây dựng từ điển xuất hiện trong training set và bài toán trở thành bài toán phân loại từ. Với bài toán này em chọn từ điển là top 5000 từ.

Ví dụ:

- Embedding vector + A -> girl
- Embedding vector + A girl -> going
- Embedding vector + A girl going -> into
- Embedding vector + A girl going into -> a.
- Embedding vector + A girl going into a -> wooden.
- Embedding vector + A girl going into a wooden -> building.

2. Mô hình chung



Hình 2: Mô hình chung

- Từ ảnh đưa qua một pre-trained model (Inception v3) lấy ra vector(CNN)
- Từ câu miêu tả là input đầu vào của mạng RNN/LSTM/GRU tạo ra vector(GRU)

Nối 2 vector này với nhau và đi qua cơ chế Attention để dự đoán được từ tiếp theo.

Chương 3. Tập Dataset

1. Bộ dữ liệu sử dụng

Bộ dữ liệu em sử dụng trong bài toán này là bộ dữ liệu MS COCO, gồm có hơn 80.000 ảnh, mỗi ảnh có ít nhất 5 chú thích phụ đề khác nhau. Nhưng do về hạn chế về tài nguyên của máy, nên em lấy 5.000 ảnh từ bộ dữ liệu này để tạo ra 25.000 dataset.

Ngoài ra, em còn train mô hình với bộ dữ liệu 500 ảnh và 1000 ảnh để so sánh đánh giá chất lượng

2. Chia dữ liệu đào tạo và thử nghiệm

```
[ ]  len(img_name_train), len(img_name_val)

     (20008, 5004)
```

25.000 dataset chia theo tỉ lệ tập train:val = 80%:20%

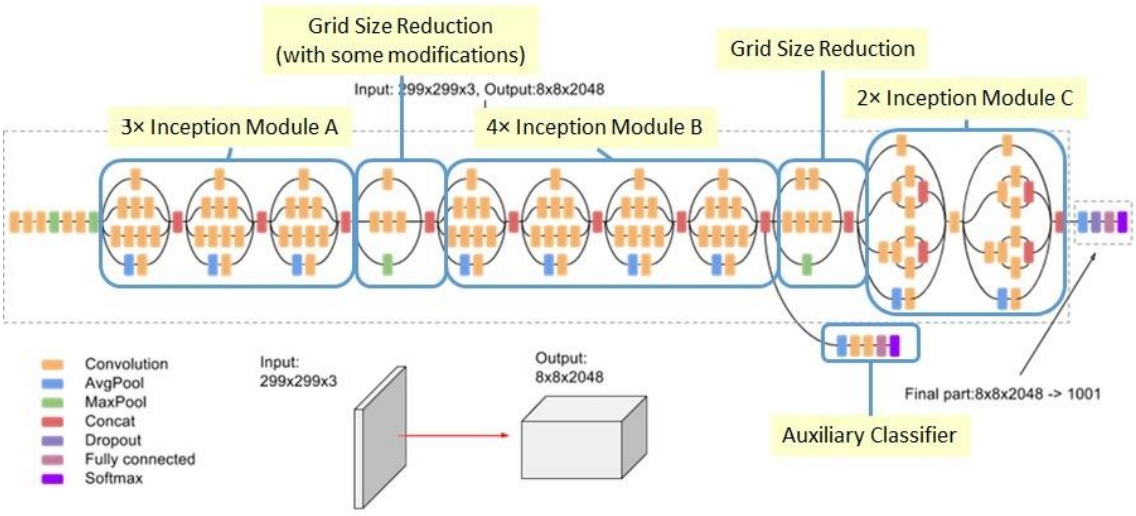
Chương 4. Chi tiết mô hình giải quyết bài toán

1. Image embedding với Inception V3.

1.1. Cấu trúc mô hình

Model Inception-v3 bằng cách xem xét lại kiến trúc, hiệu quả tính toán tốt hơn và ít tham số hơn. Với 42 layers, tỉ lệ lỗi thấp của mô hình đã giúp nó trở thành 1st Runner Up for image classification in ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2015. Thay vì trong mỗi Conv layer chỉ dùng 1 kernel size nhất định như 3*3, 5*5, thì giờ ở một layer có nhiều kernel size khác nhau, do đó mô hình có thể học được nhiều thuộc tính khác nhau của ảnh trong mỗi layer. Và đó là lý do em sử dụng Inception-v3 để lấy ra những feature của ảnh.

Đưa ảnh qua pre-trained model(inception V3) để mô hình có thể học được nhiều thuộc tính khác nhau của ảnh



Hình 3: Model Inception V3

Input pretrained model: 299x299x3

Output pretrained model: 8x8x2048

Đầu tiên, để sử dụng được model bạn sẽ chuyển đổi hình ảnh sang định dạng mong đợi của InceptionV3 bằng cách:

- Thay đổi kích thước hình ảnh thành 299px x 299px.
- Xử lý trước hình ảnh bằng phương thức preprocess_input để chuẩn hóa hình ảnh sao cho nó chứa các pixel trong phạm vi từ -1 đến 1, phù hợp với định dạng của hình ảnh được sử dụng để huấn luyện InceptionV3.

Model InceptionV3 bao gồm những lớp Convolution, Pooling, Dropout, Fully connected, 3 Inception modun A, 4 Inception modun B, 2 Inception modun C, đầu ra phụ (auxiliary classifier) và 2 cấu trúc giảm chiều.

1.2. Tổng quan

- Inception-v3 có 42 layers với 24 triệu tham số
- Toàn bộ layer tích chập theo sau bởi 1 layer batch normalization và 1 ReLU activation giúp cho quá trình huấn luyện nhanh hơn và tránh over-fitting
- Mô hình sử dụng cơ chế Label smoothing

$$new\_labels = (1 - \varepsilon) * one\_hot\_labels + \frac{\varepsilon}{K}$$

Trong đó: ε là siêu tham số, thường được chọn bằng 0.1

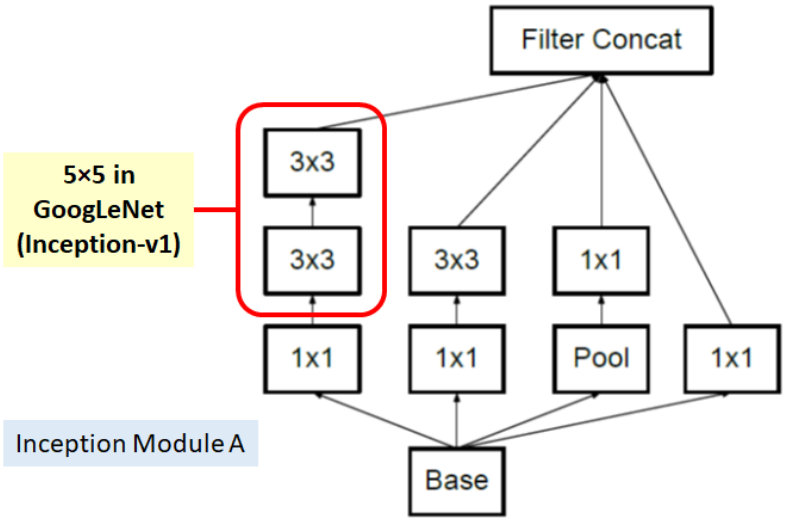
K là số class

Mục đích ngăn logit lớn nhất lớn hơn nhiều so với logit còn lại. Bởi vì trong thực tế caption cho ảnh có thể không được chuẩn. Việc này giúp cho mô hình tránh bị over-fitting.

1.3. Factorizing Convolutions

Mục đích làm giảm số lượng tham số mà không làm giảm hiệu quả mạng.

1.3.1. Inception module A



Hình 4: Inception Module A

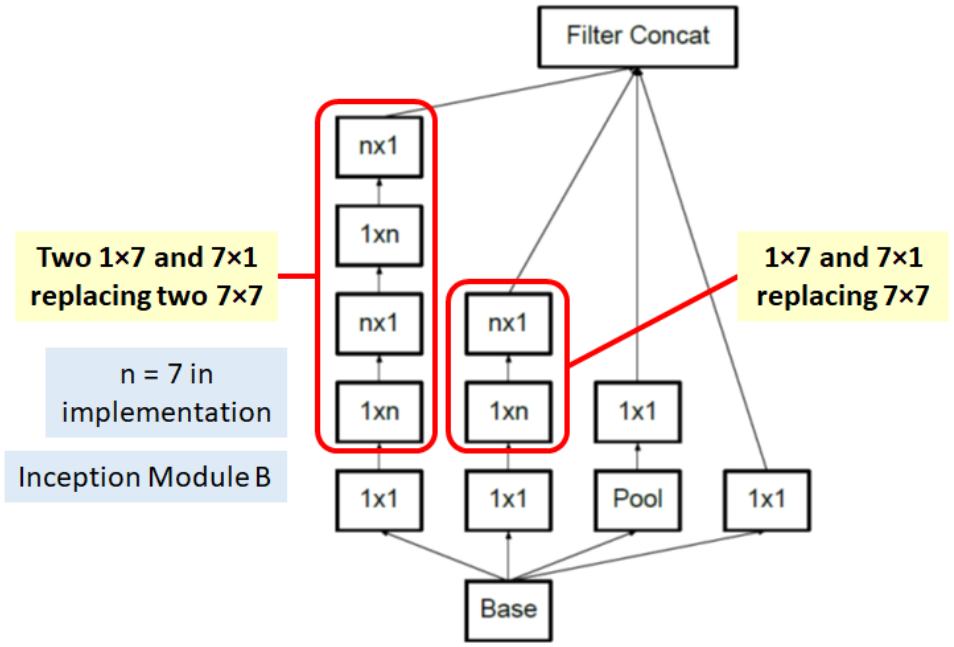
So với InceptionV1 thì InceptionV3 thay thế hai 3x3 convolution cho một 5x5 convolution.

Nếu sử dụng một layer 5x5: số lượng tham số = 5x5 = 25

Nếu sử dụng hai layer 3x3: số lượng tham số = 3x3 + 3x3 = 18

Số lượng tham số giảm khoảng 28%.

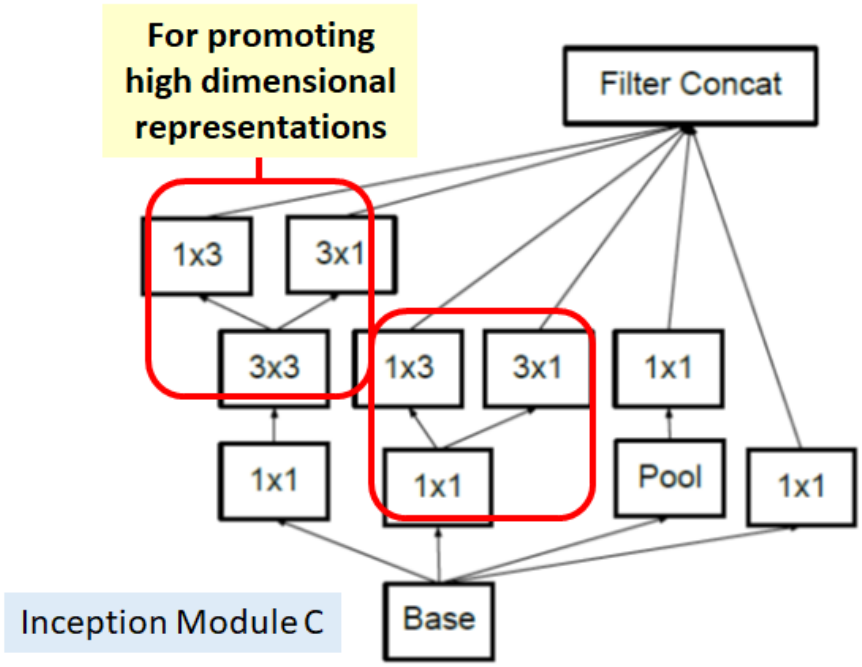
1.3.2. Inception module B



Hình 5: Inception Module B

Inception Module B cải tiến của Inception Module A: thay tích chập 3x3 bằng tích chập 7x7 ở nhánh 1 và 2. Phân tích tổ chập 7x7 thành 2 tích chập con liên tiếp 7x1 và 1x7.
Số lượng tham số của mô hình khi sử dụng 7x1 và 1x7 thay cho 7x7 đã giảm từ 49 xuống còn 14.

1.3.3. Inception module C



Hình 6: Inception module C

Inception module C cải tiến từ module B: Thay tích chấp 7x1 thành 3x1 và 1x7 thành 1x3 và đặt song song với nhau.
Việc này giúp cho mô hình train nhanh hơn và số lượng tham số giảm từ 14 còn 6.

1.3.4. Cấu trúc giảm chiều hiệu quả

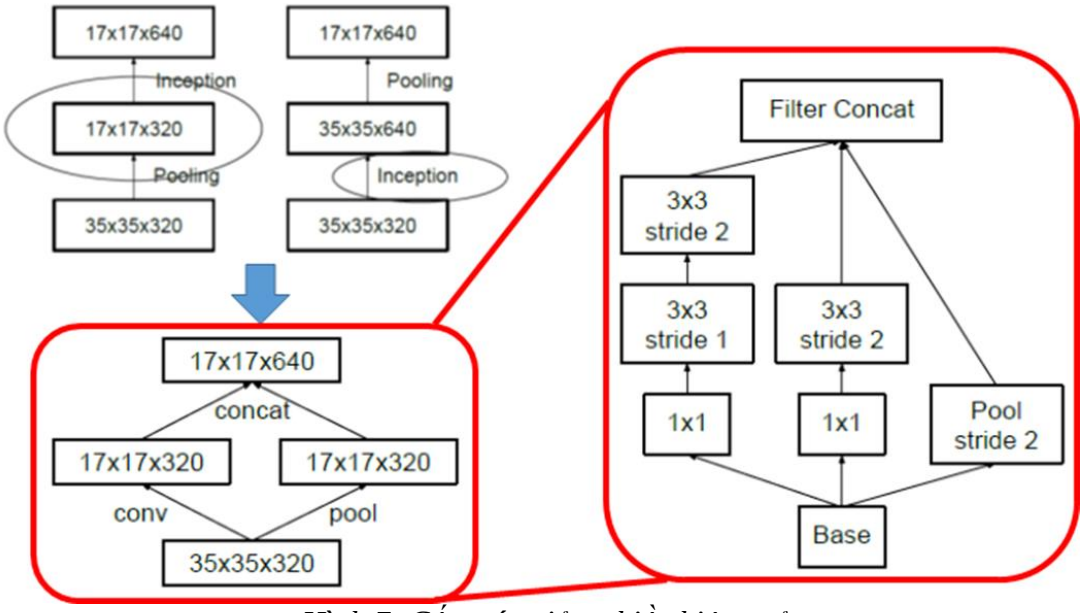
Nếu giảm chiều theo các truyền thống thì chúng ta phải xếp chồng các pooling layer và convolution layer với nhau, việc này tốn rất nhiều tham số và thời gian training mô hình dài.

Ví dụ muốn giảm chiều từ tensor 35x35x320 về 17x17x640 thì chúng ta có hai cách.
Cách 1: từ tensor 35x35x320 pooling về tensor 17x17x640 và tiếp tục cho qua convolution layer trở thành tensor 17x17x640. Cách này sẽ khiến chúng ta tốn $640*320*n*n = 204800n^2$ tham số với n là chiều của kenel

Cách 2: từ tensor 35x35x320 ta cho qua lớp Convolution layer để ra tensor 35x35x640 và tiếp tục cho qua pooling layer trở thành tensor 17x17x640. Cách này khiến chúng ta tốn $640*640*n*n = 409600n^2$ tham số.

Mạng Inception-V3 đề xuất cách giảm chiều hiệu quả hơn bằng cách từ tensor 35x35x320 một đường đi qua convolution layer để đưa ra tensor 17x17x320 và một đường đi qua pooling layer để giảm chiều về tensor 17x17x320. Rồi nối 2 tensor với nhau bằng lớp filter concat để được tensor 17x17x640. Cách làm này giúp chúng ta chỉ cần $320*320*n*n = 104400n^2$ tham số.

Mô hình bên dưới đã thực hiện hóa ý tưởng này.



Hình 7: Cấu trúc giảm chiều hiệu quả

2. Preprocessing text

- Đầu tiên ta xử lý text qua một số bước cơ bản:
  - Chuyển chữ hoa thành chữ thường, ví dụ “Hi” -> “hi”
  - Loại bỏ những kí tự đặc biệt như “!@#%...”
  - Loại bỏ những từ có chữa số.

- Thêm “strat” và “end” vào đầu và cuối của mỗi caption để đánh dấu sự bắt đầu và kết thúc của câu.
- Do các câu có độ dài ngắn khác nhau nên chúng ta phải thêm padding để độ dài các câu bằng câu dài nhất. Do đó kích thước từ điển trở thành 5001 từ (1 từ để padding).

3. Word embedding

- Em xây dựng từ điển gồm top 5001 từ. Bởi vì ta không quan tâm lắm tới những từ chỉ xuất hiện 1 vài lần, nó giống như nhiễu vậy và không tốt cho việc học và dự đoán từ của model. Thay thế tất cả các từ khác bằng mã thông báo "UNK" (không xác định). Thứ tự từ càng nhỏ thì số lần xuất hiện trong câu càng lớn.

<unk>	1
'vietnamese'	2
'i'	3
'am'	4
'people'	5
'are'	6
'pretty'	7
'friendly'	8

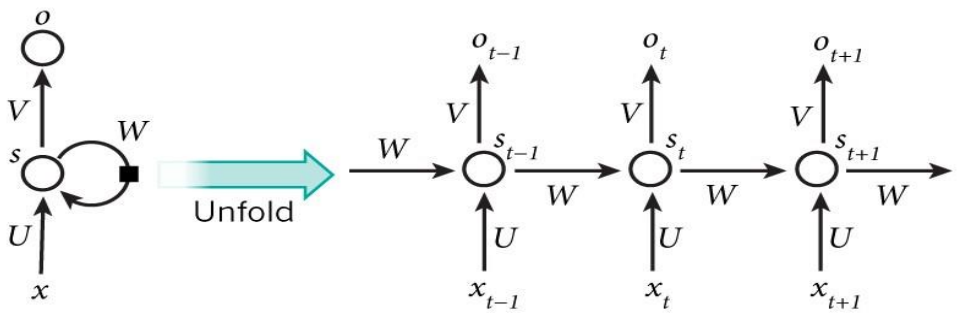
Em chuyển câu thành vector bằng cách text to sequence:

Ví dụ với câu: “= ['I am Vietnamese people','Vietnamese people are pretty friendly',]”

Chuyển thành [[3, 4, 2, 5, …], [2, 5, 6, 7, 8, …]]

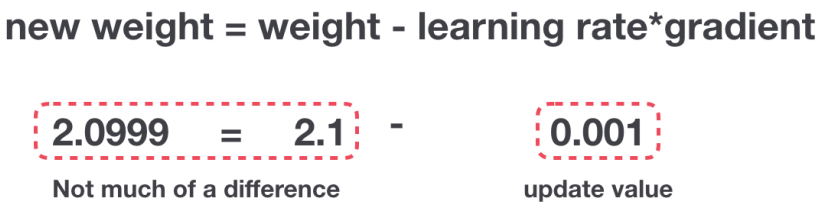
4. GRU

Kiến trúc Recurrent Neural Network (RNN) được sinh ra để giải quyết các bài toán có dữ liệu tuần tự. RNN đưa thông tin về trạng thái ẩn (hidden state) được tính toán khi kết quả của quá khứ thành đầu vào cho quá trình hiện tại. Đây là cơ chế giúp RNN xử lí tuần tự: các từ phía trước sẽ ảnh hưởng đầu ra của từ phía sau.

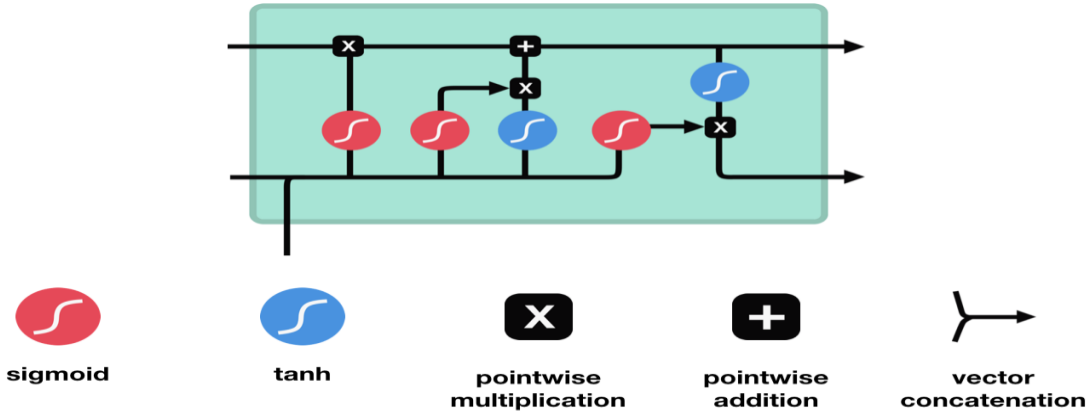


Hình 8: Cấu trúc RNN

Tuy vậy, do kiến trúc của nó khá đơn giản nên khả năng liên kết các thành phần có khoảng cách xa trong câu không tốt. Vì thế, nếu bạn đang xử lý một đoạn văn dùng RNN, nó có thể bỏ qua những chi tiết ở đầu đoạn văn đó do bộ nhớ có hạn. Điểm yếu của RNN chính là vanising gradient. Khi câu dự đoán ra quá dài có nghĩa là mạng sẽ sâu. Khi backpropagation lại thì những layer đầu tiên có weight không thay đôủ nhiều và mô hình gần như không học được gì.



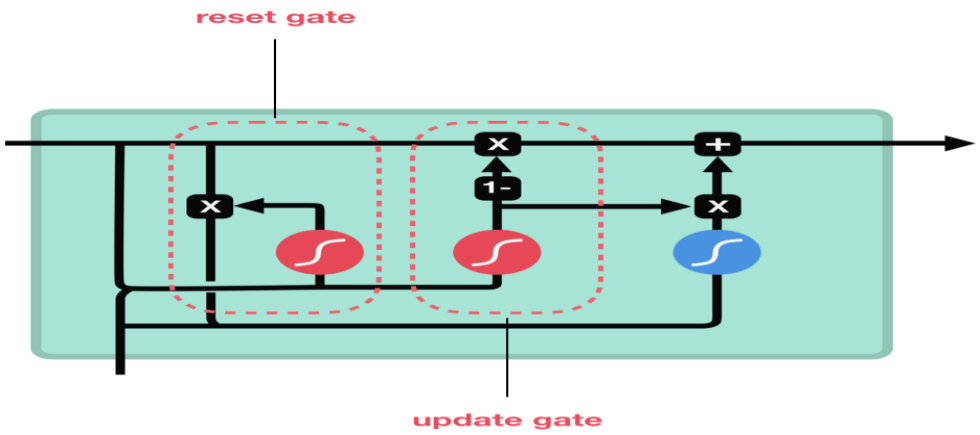
LSTM có cơ chế hoạt động giống RNN nhưng có 1 số thay đổi trong trạng thái



Hình 9: Cấu trúc LSTM

LSTM có 3 cổng: cổng đầu ra, cổng đầu vào, cổng quên. Sự khác biệt ở đây là cổng quên và trong mỗi trạng thái có thêm hàm sigmod chiếu về (0,1) để dễ dàng xác định được các phần thông tin quan trọng và bỏ qua được những thông tin không cần thiết. LSTM tạo ra để tránh hiện tượng vanishing gradient.

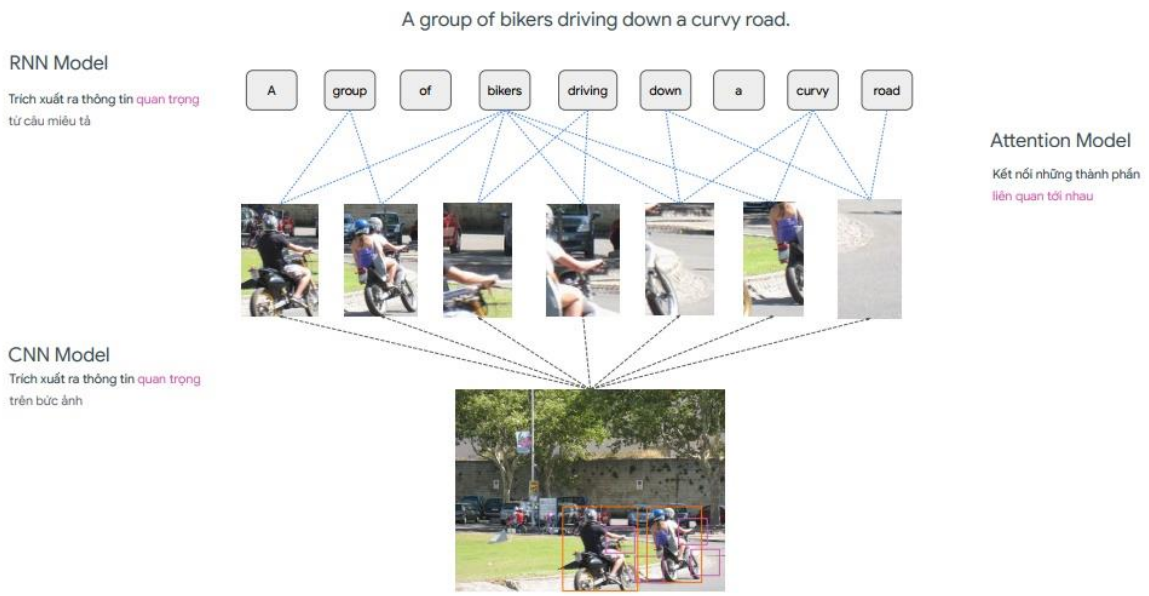
GRU chỉ có 2 cổng để chắt lọc thông tin, cổng reset và cổng update. Nó không có trạng thái cell mà chỉ có đầu ra vừa dùng để đưa ra quyết định vừa dùng để thông tin cho các bước tiếp theo



Hình 10: Cấu trúc GRU

Cả 2 cổng đều có tác dụng chắt lọc thông tin từ các đầu vào của cell để đưa ra một output thỏa mãn cả 2 tiêu chí là lưu giữ thông tin quá khứ và có khả năng đưa ra quyết định hiện tại một cách chính xác nhất. Do giảm bớt tính phức tạp nên GRU hoạt động nhanh hơn đôi chút so với LSTM. Về tính hiệu quả thì rất khó để đưa ra kết luận chính xác, nhưng có thể nói mức độ phổ biến LSTM là vượt trội khi so với GRU. Em sử dụng mạng GRU để tránh vanishing gradient và giúp mô hình train nhanh hơn.

5. Kết hợp mô hình



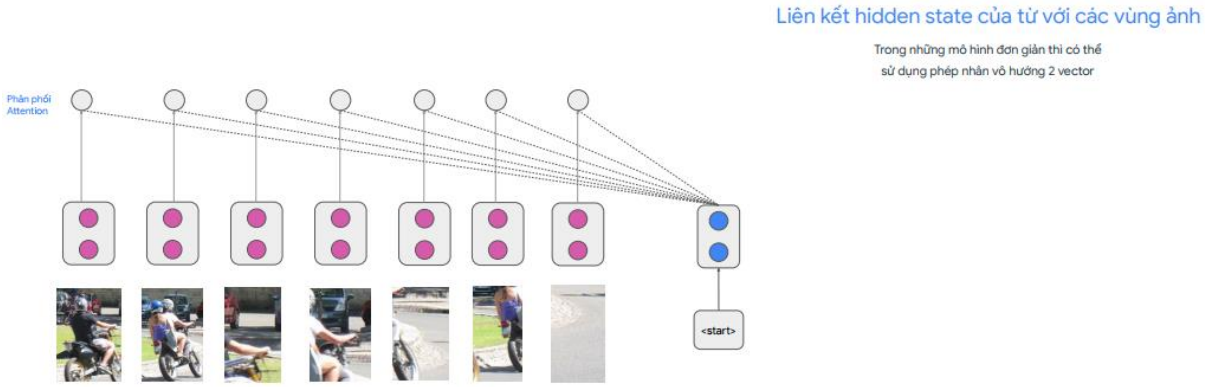
Hình 11: Kết hợp các mô hình

- Đầu tiên ta đưa ảnh qua model InceptionV3 để lấy ra những đặc trưng của ảnh, cho ra một tensor có hình dạng (8, 8, 2048).
- Tensor này sau đó được chuyển qua Encoder CNN (là một mạng neural network) để học những thuộc tính một cách cô đọng hơn. Output của encoder là tensor (8, 8, 256).
- RNN (ở đây là GRU) sẽ trích xuất ra những thông tin quan trọng của từ miêu tả.
- Sử dụng cơ chế Attention để kết hợp đầu ra của decoder và các vector đặc trưng của ảnh để dự đoán ra từ tiếp theo.

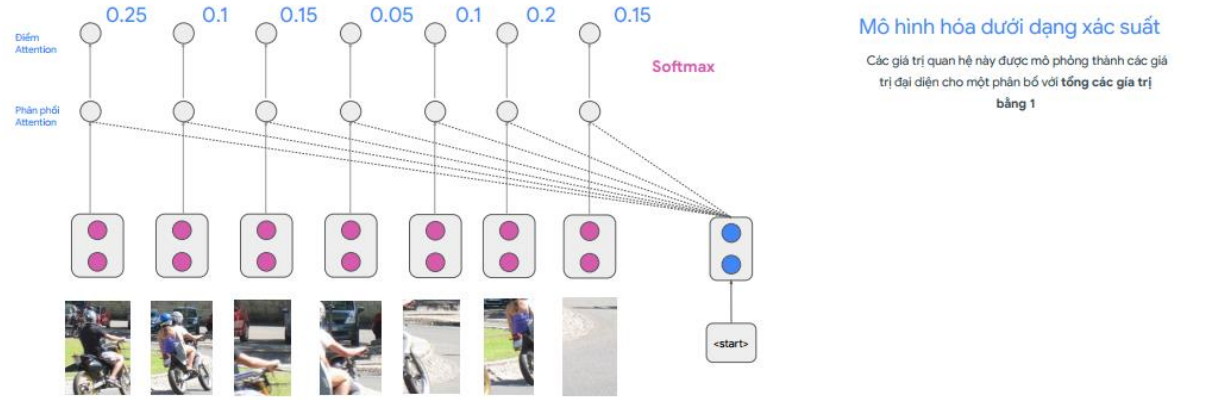
6. Cơ chế Attention

Sử dụng vector để thực hiện kết nối đầu ra của decoder với những khu vực ảnh quan trọng (feature từ đầu ra của CNN). Cơ chế attention không chỉ dùng context vector mà còn sử dụng hidden state ở từng từ trong input với trọng số ảnh hưởng tương ứng, nên việc dự đoán từ tiếp theo sẽ tốt hơn cũng như không sợ tình trạng từ ở xa bị mất thông tin ở context vector.

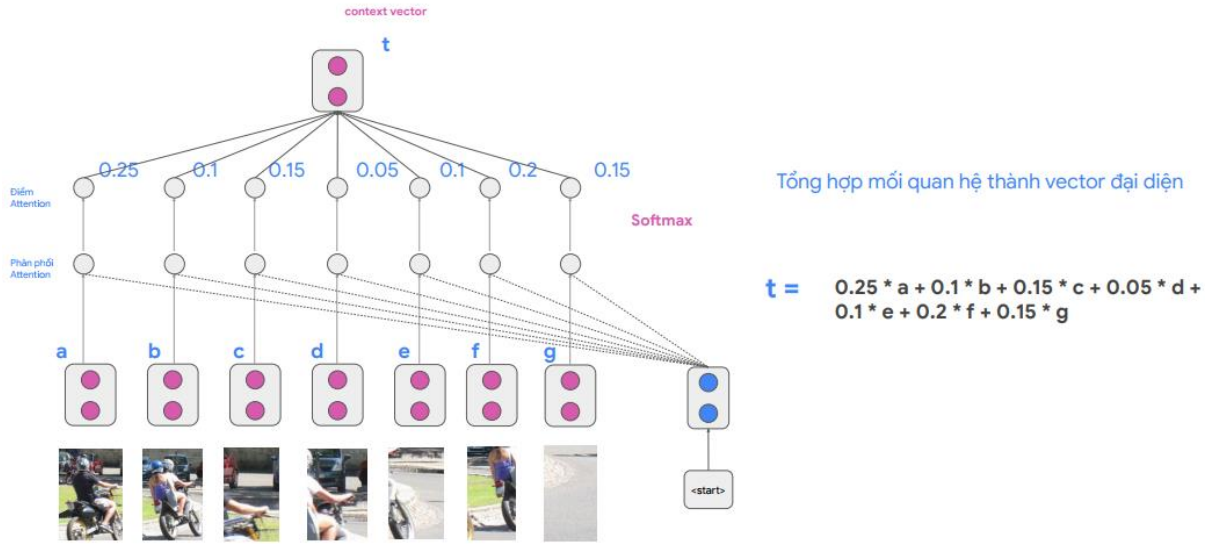
Ngoài ra các mô hình deep learning hay bị nói là hộp đen (black box) vì mô hình không giải thích được, attention phần nào giúp visualize được kết quả dự đoán, ví dụ từ nào ở ouptut ảnh hưởng nhiều bởi vùng ảnh nào trong input. Do đó model học được quan hệ giữa các vùng ảnh trong input và từ trong output để đưa ra kết quả dự đoán.



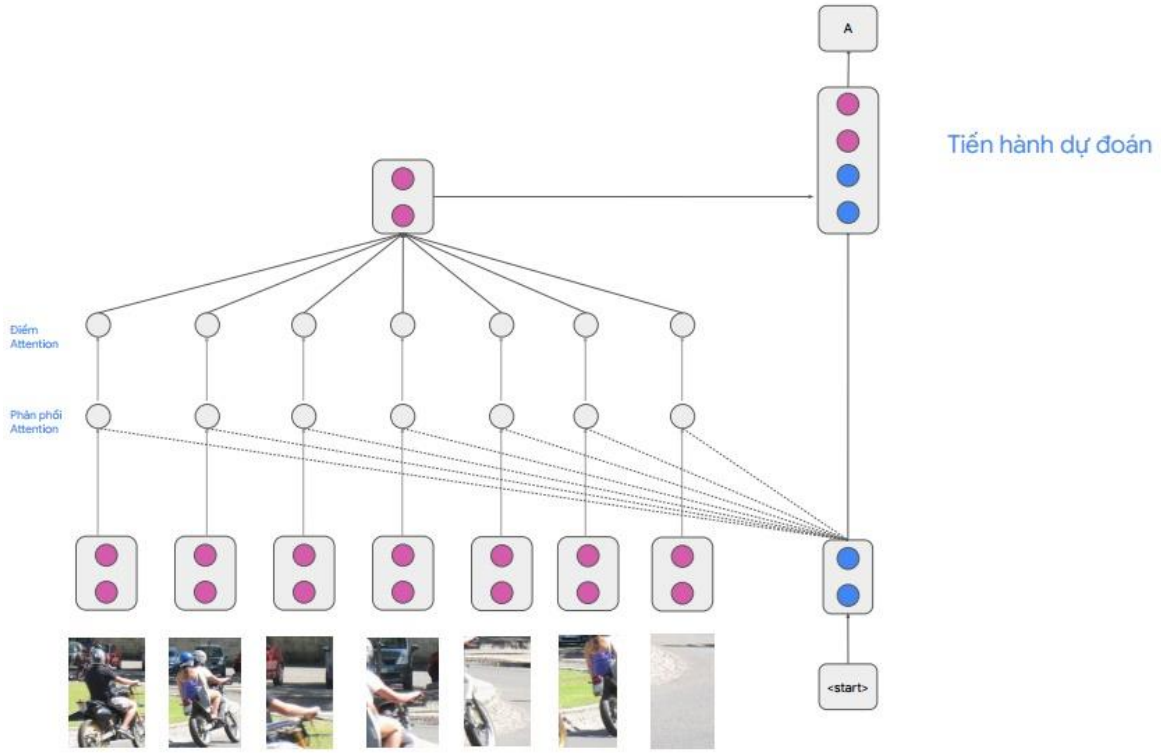
Hình 12: Liên kết hidden state của từ với từng vùng ảnh



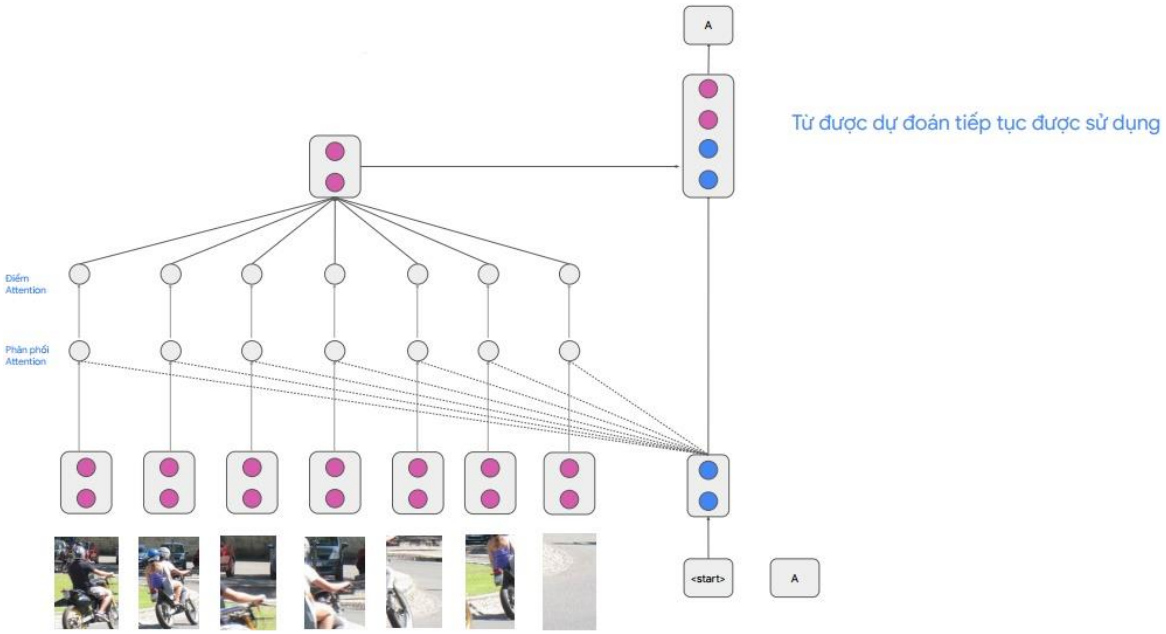
Hình 13: Mô hình hóa dạng xác suất



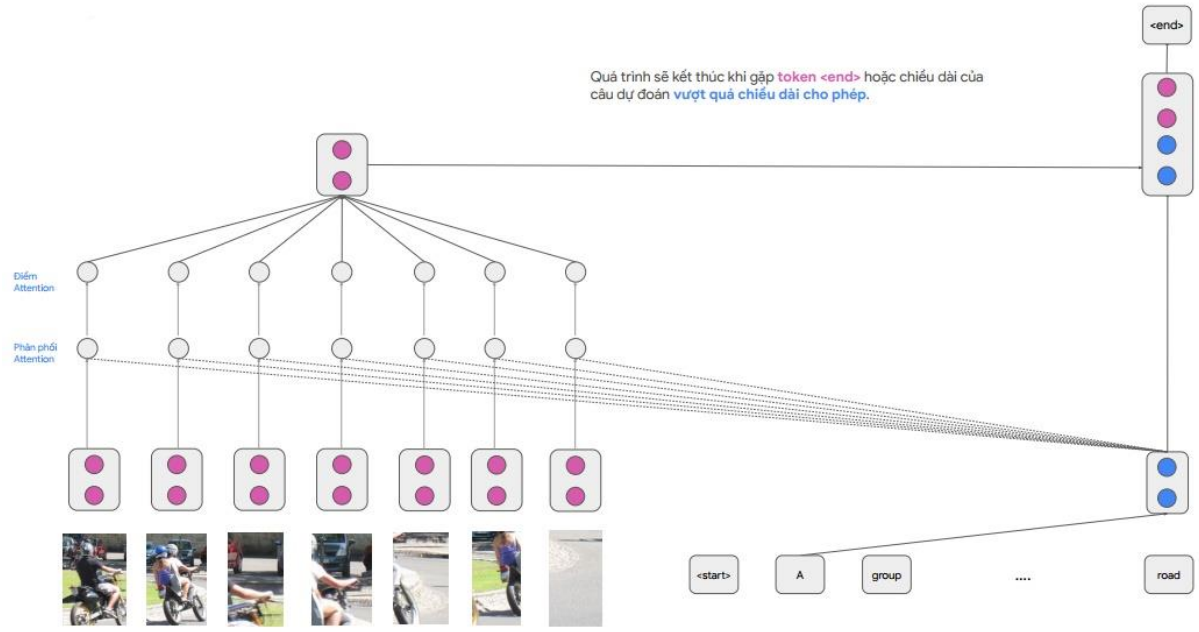
Hình 14: Tổng hợp thành vector đại diện



Hình 15: Tiến hành dự đoán



Hình 16: Từ tiếp theo được sử dụng

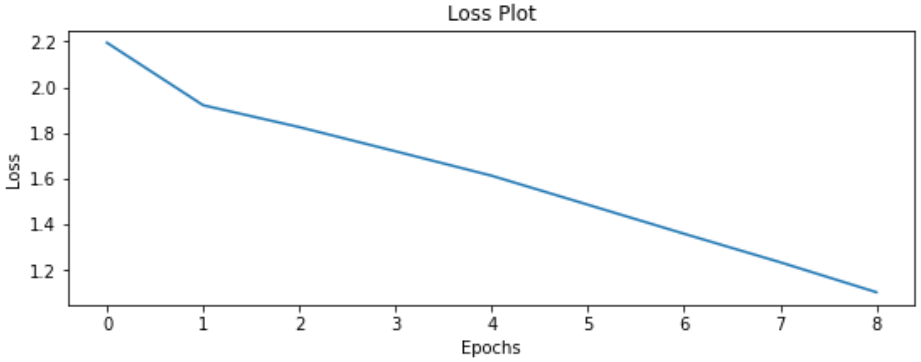


Hình 17: Dự đoán kết thúc

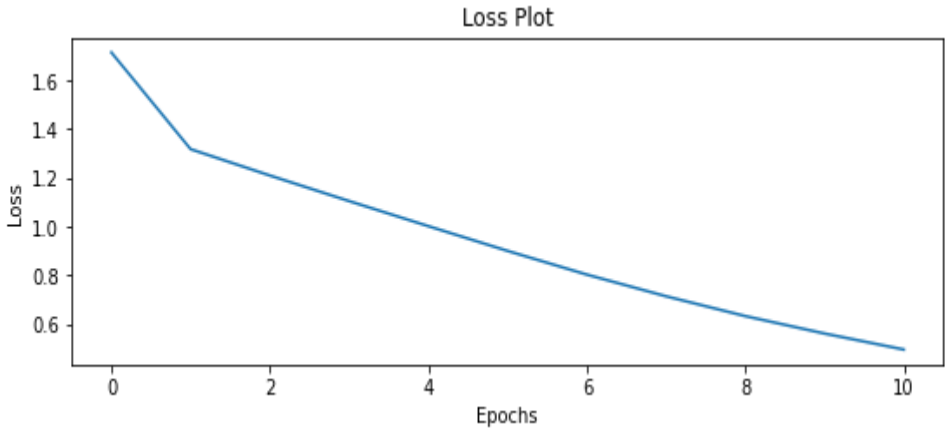
7. Training

- Bài toán là dự đoán ra từ tiếp theo trong chuỗi ở input với ảnh hiện tại, nên output là một trong số 5001 từ ta có.
- Chiến thuật tối ưu em sử dụng là Adam optimizer.
- Ở bài toán này, do là bài toán phân loại nên softmax activation và hàm đánh giá lỗi là hàm categorical_CrossEntropy được sử dụng.

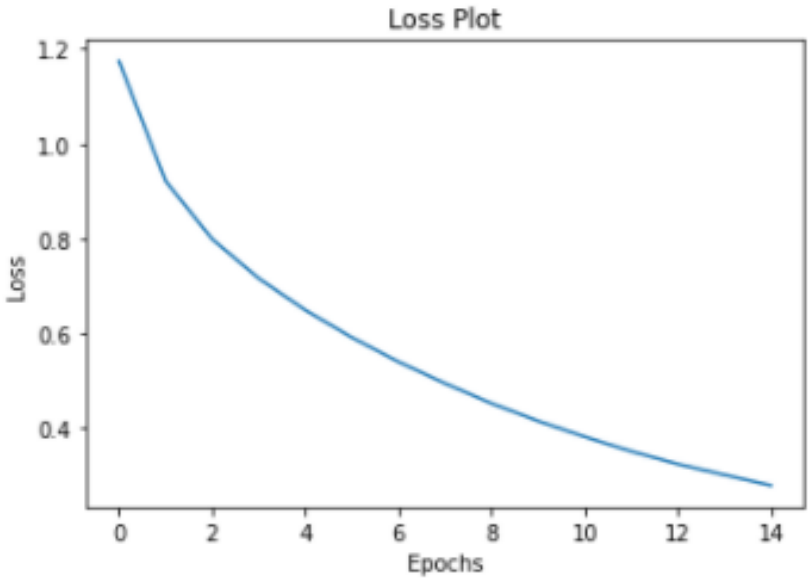
8. Kết quả và dự đoán



Hình 18: Loss 500 ảnh



Hình 19: Loss 1000 ảnh

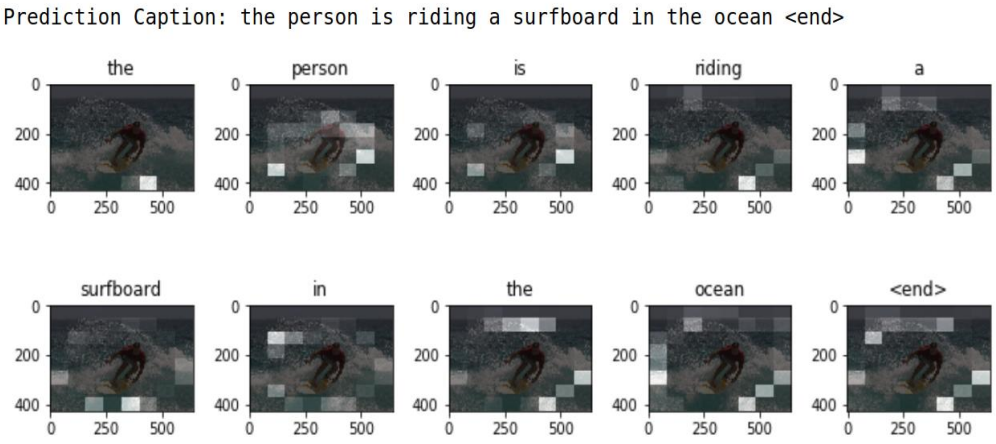


Hình 20: Biểu đồ loss với 5000 ảnh

⇨ Từ 3 biểu đồ trên chúng ta có thể thấy càng nhiều dữ liệu thì mô hình hội tụ càng nhanh và tốt hơn.

- BLEU score trên tập test = 0.51.
- Ví dụ ta có ảnh đầu vào bên dưới

Hình 20: Hình ảnh thử nghiệm



Hình 21: Kết quả thử nghiệm

- Kết luận: Mô hình nhận diện những vật thể khá là tốt, tuy nhiên chưa chuẩn lắm về ngữ cảnh của ảnh và ngữ pháp của câu.

9. Khó khăn và hướng phát triển tương lai

- Trong quá trình làm bài toán em gặp những khó khăn và cách giải quyết:
    - Do mô hình học sâu lớn nên mô hình dễ bị overfitting. Em nhiều lần siêu tham số epochs để dừng lại trước khi mô hình bị overfit.
    - Em còn gặp phải khó khăn về tài nguyên để train mô hình. Do tài nguyên em sử dụng có hạn là google colab nên lúc chỉnh sửa và tối ưu các siêu tham số của mô hình mất nhiều thời gian và công sức. Em cũng đã cắt bớt bộ dữ liệu giúp cho thời gian train nhanh hơn và mô hình có độ tốt vừa đủ.

- Hướng phát triển tương lai:
    - Em sử dụng text to sequence để word embedding dữ liệu text. Thay vì cách này thì em sẽ thử cách word2vec (embedding câu bằng ngữ cảnh) để nâng cao hiệu quả của mô hình.
    - Thay vì tự code attention và decoder thì em sẽ tìm thêm những pretrained model để có thể tách đặc trưng của câu tốt hơn nhằm nâng cao khả năng dự đoán từ.
    - Và tất nhiên, một trong những cách cơ bản để cải thiện mô hình hoc sâu là em sẽ tìm thêm dữ liệu và train mô hình với bộ dữ liệu lớn hơn.

Tài liệu tham khảo

[1] Ứng dụng thêm mô tả cho ảnh? https://nttuan8.com/bai-15-ung-dung-them-mo-ta-cho-anh-image-captioning/

[2] https://www.tensorflow.org/tutorials/text/image_captioning

[3] https://sh-tsang.medium.com/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c

[4] Show, Attend and Tell: Neural Image Caption Generation with Visual Attention