ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ

VI ANH QUÂN

TRẦN ANH MINH

NGUYỄN NHẬT TÙNG

NGUYỄN TIẾN ĐẠT

MÔ HÌNH SINH TẢ ẢNH DỰA TRÊN CẢM XÚC

Báo cáo kết thúc môn Machine Learning

Ngành Kĩ thuật Điện tử và Tin học

(Chương trình đào tạo chuẩn)

Hà Nội - 2023

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ

VI ANH QUÂN
TRẦN ANH MINH
NGUYỄN NHẬT TÙNG
NGUYỄN TIẾN ĐẠT

MÔ HÌNH SINH TẢ ẢNH DỰA TRÊN CẢM XÚC

Báo cáo kết thúc môn Machine Learning
Ngành Kĩ thuật Điện tử và Tin học
(Chương trình đào tạo chuẩn)

Giảng viên hướng dẫn: TS. Phạm Tiến Lâm

Hà Nội - 2023

Lời cảm ơn

Trong quá trình học tập môn Machine learning, chúng em đã nhận được sự giúp đỡ nhiệt tình của thầy Phạm Tiến Lâm và thầy Đặng Văn Báu. Thầy đã cung cấp cho chúng em những kiến thức vô cùng quan trọng để phục vụ trong quá trình học tập và nghiên cứu. Chúng em rất trân trọng khoảng thời gian cùng thầy nghiên cứu môn học vô cùng thú vị này.

Do những hạn chế về kiến thức, dự án này của chúng em cũng không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự đóng góp, phê bình của các thầy để đề tài của chúng em được hoàn thiện hơn. Cuối cùng em xin kính chúc thầy nhiều sức khỏe, thành công và hạnh phúc.

Chúng em xin chân thành cảm ơn!

Hà Nội, tháng 6 năm 2023

Sinh viên

Mục lục

Danh sách hình vẽ

Chương 1 MỞ ĐẦU

Hiện nay trí tuệ nhân tạo là một xu hướng công nghệ trên toàn thế giới. Nó đã xuất hiện ở mọi nơi như là mạng xã hội, thiết bị điện tử, xe cộ, ... Xuất phát từ nhu cầu ngày càng cao của con người về sự tiện lợi, thông tin nhanh chóng thì mô hình nhận diện hành động con người cũng đã được ra đời.

Nhận diện hành động con người ở ảnh tĩnh có rất nhiều tiềm năng ứng dụng vào đời sống như là thiết bị hỗ trợ người khiếm thị, gán nhãn tự động, ... Một giải pháp trực tiếp cho vấn đề đó chính là từ hình ảnh sinh ra được văn bản mô tả chính xác ảnh đó với lời văn hợp lý. Tuy nhiên thử thách được đặt ra đó là con người có rất nhiều trạng thái hoạt động phức tạp khác nhau cùng với ngữ pháp mỗi loại ngôn ngữ là khác nhau. Chính vì vậy bài báo cáo này sẽ tập trung vào một số trạng thái cơ bản của con người và mô tả nó bằng tiếng Anh. Các mô hình Machine learning và Deep learning được sử dụng trong bài báo cáo này là CNN, RNN và LSTM. Đây là các mô hình rất phổ biến trên thế giới, được tin dùng trong việc xử lí ảnh và ngôn ngữ tự nhiên.

Chương 2 TỔNG QUAN

2.1 Dataset sử dụng

Việc sử dụng nhiều dataset khác nhau cho phép đa dạng dữ liệu đầu vào và kiểm tra được tính linh hoạt của mô hình trong nhiều trường hợp khác nhau.

2.1.1 Emotional attitude captioning dataset

Dataset sử dụng trong bài có 3840 ảnh khác nhau, được lấy từ IMDB dataset. Mỗi ảnh được gán nhãn với một số trạng thái như là "laughing", "flirting", "kissing", "angry", "happy", ...[1].

Hình 2.1: Gán nhãn: Two guys are looking on each other.

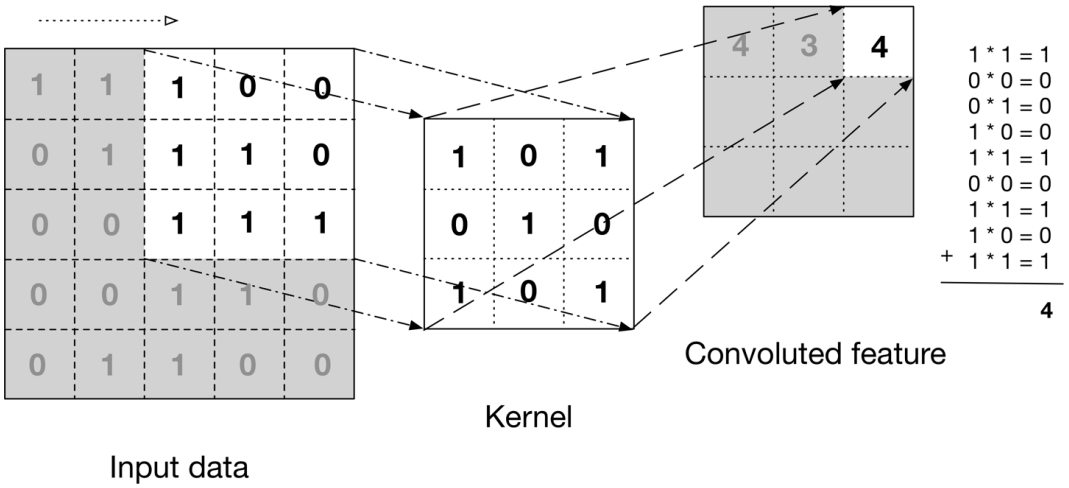
2.1.2 Flickr8

Dataset được lấy trên trang web Kaggle [2].

2.2 Mô hình CNN

CNN (Convolutional Neural Network) là một mô hình Deep learning để xử lí dữ liệu ở dạng lưới như là hình ảnh, được thiết kế để học một cách tự động và thích ứng với hệ thống phân cấp không gian của các tính năng, từ pattern thấp tới cao.[3]

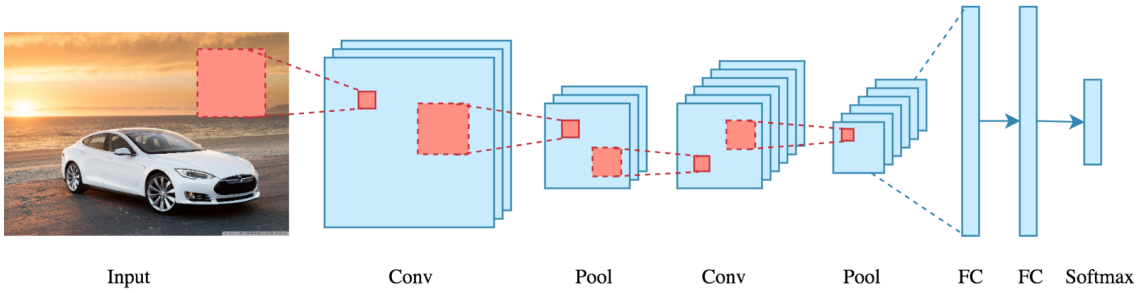
Nó sử dụng một kĩ thuật đặc biệt gọi là tích chập (Convolution), một phép toán trên hai phương trình để đưa ra phương trình thứ ba mô tả cách hình dạng của một phương trình bị thay đổi bởi phương trình còn lại.[4]



Hình 2.2: Tích chập.

Cấu trúc của CNN thường gồm ba loại lớp:

- Convolution
- Pooling
- Fully connected



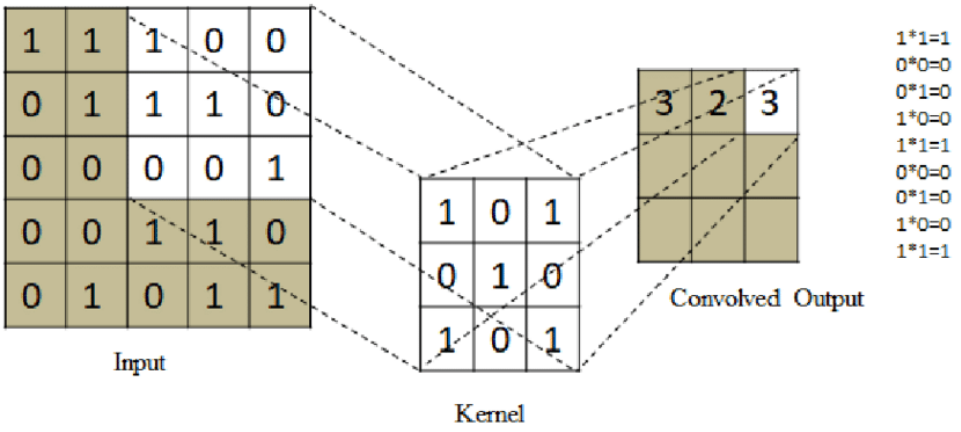
Hình 2.3: Các lớp của mô hình CNN.

Khi đầu vào là một ảnh, mỗi lớp sẽ tạo ra các hàm kích hoạt sau đó tiếp tục được đưa vào lớp tiếp theo. Lớp đầu tiên thường sẽ trích xuất ra đặc điểm cơ bản của ảnh như các cạnh ngang và chéo. Đầu ra này sẽ được đưa vào lớp tiếp theo để xác định các góc và cạnh tổ hợp. Càng đưa vào nhiều lớp sâu hơn, ta càng tìm được các đặc điểm phức tạp như vật thể, mặt, ...[4].

2.2.1 Convolutional layer

a) Kernel

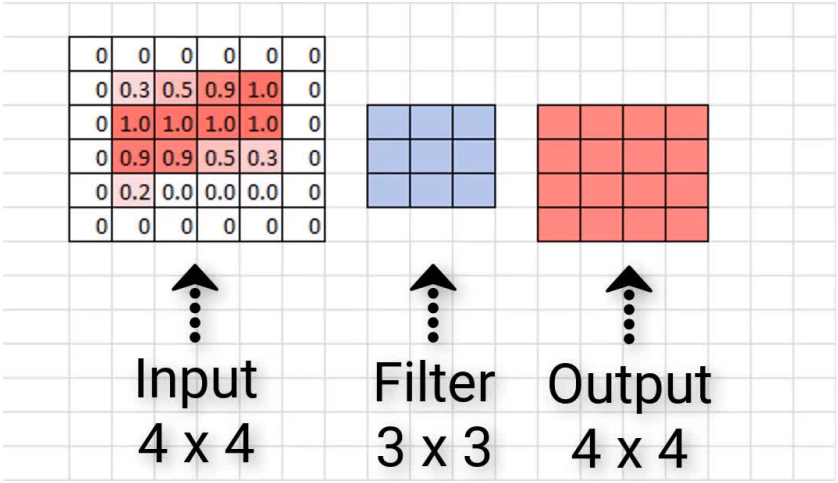
Kernel là một ma trận vuông có kích thước KxK với K là các số lẻ. Ma trận X tích chập với kernal sẽ được ma trận Y.



Hình 2.4: Phép tính với kernel.

b) Padding

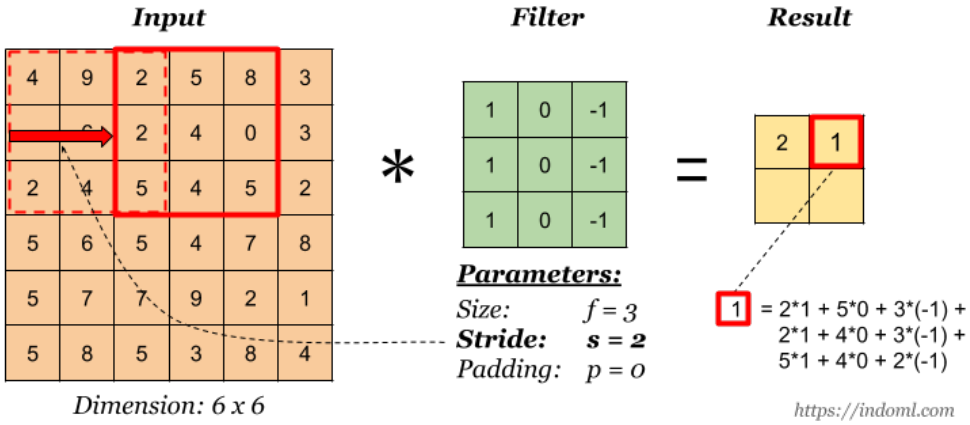
Sau phép biến đổi Convolution, ma trận Y sẽ có kích thước nhỏ hơn X. Nếu muốn ma trận Y có cùng kích thước với X thì ta cần thêm giá trị 0 vào viền ngoài ma trận X trước khi thực hiện phép tính. Việc thêm viền ngoài này được gọi là Padding.



Hình 2.5: Thêm padding.

c) Stride

Nếu thực hiện phép tích chập lần lượt trên các phần tử $x_{i,j}$, $x_{i,j+1}$, ... cho đến hết ma trận thì khi đó Stride là 1. Stride bằng n thì ta sẽ thực hiện phép tính trên các phần tử $x_{i,j}$, $x_{i,j+n}$, ...

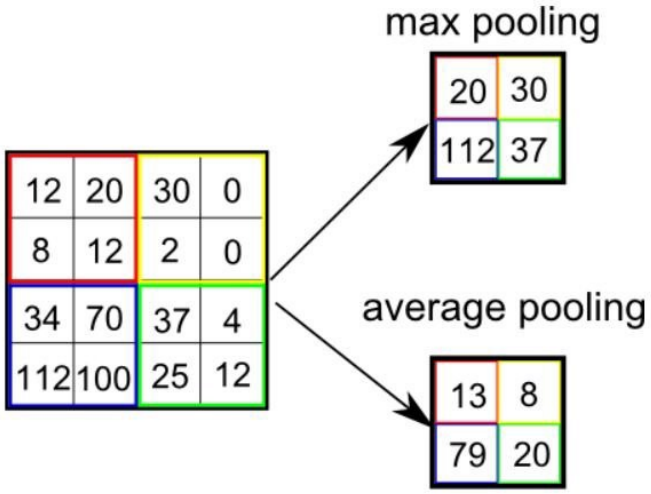


Hình 2.6: Stride = 2.

2.2.2 Polling layer

Pooling layer thường được dùng giữa các convolutional layer, để giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Việc giảm kích thước dữ liệu giúp giảm các phép tính toán trong model.

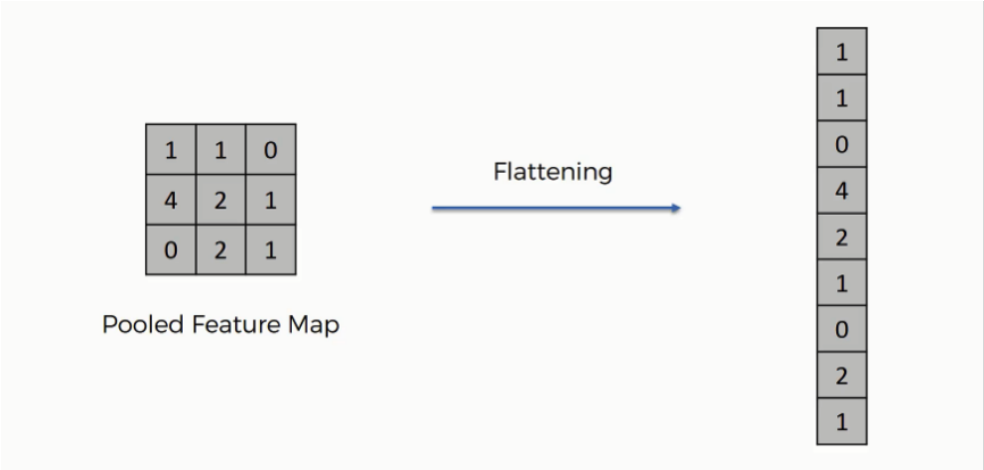
Bên cạnh đó, với phép pooling kích thước ảnh giảm, do đó lớp convolution học được các vùng có kích thước lớn hơn. Ví dụ như ảnh kích thước 224*224 qua pooling về 112*112 thì vùng 3*3 ở ảnh 112*112 tương ứng với vùng 6*6 ở ảnh ban đầu. Vì vậy qua các pooling thì kích thước ảnh nhỏ đi và convolutional layer sẽ học được các thuộc tính lớn hơn.



Hình 2.7: Hai loại polling.

2.2.3 Fully connected layer

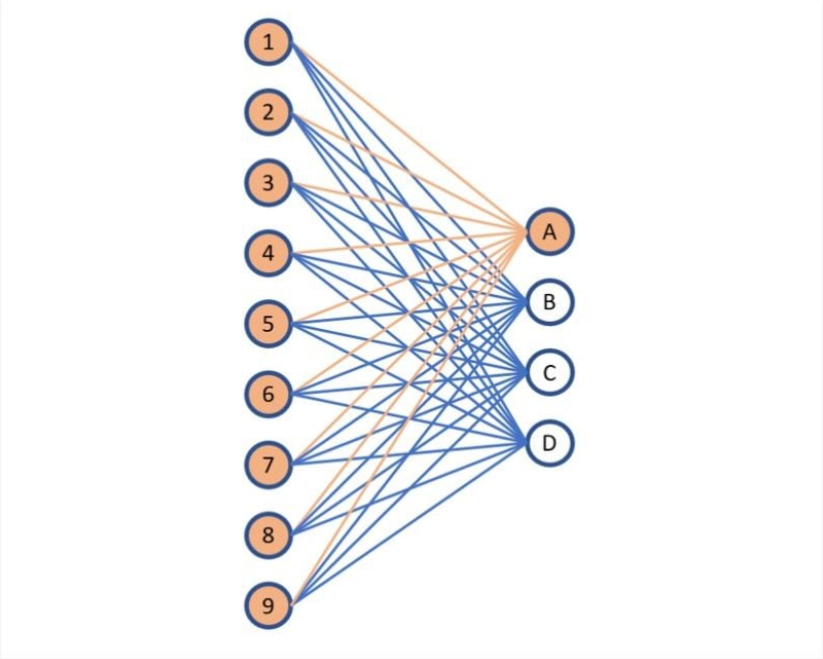
Sau khi qua Convolution layers và Polling layers thì đầu ra vẫn là một ma trận nhiều hàng nhiều cột. Trước khi đưa dữ liệu vào lớp Fully connected ta cần phải biến đổi dữ liệu thành dạng nhiều hàng một cột.



Hình 2.8: Flattening.

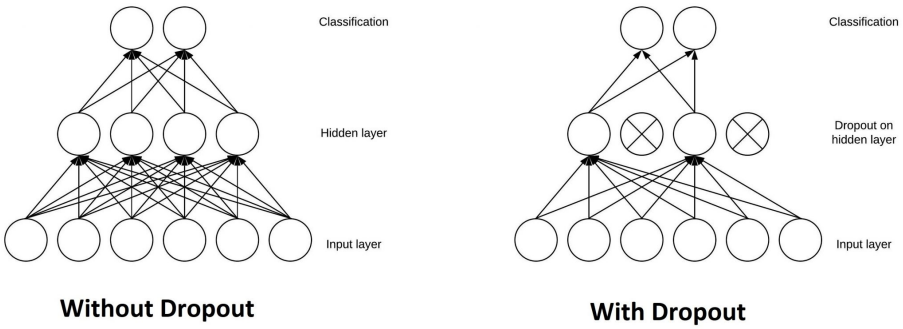
Lớp Fully connected đề cập tới mạng nơ-ron trong đó mỗi nơ-ron áp dụng một phép biến đổi tuyến tính cho vecto đầu vào thông qua ma trận trọng số. Kết quả là tất cả đầu vào của vecto đầu vào đều có ảnh hưởng đến tất cả đầu ra của vecto đầu ra [5].

Trong quá trình đào tạo, để tránh mô hình bị overfitting (mô hình dự đoán có kết quả chính xác rất cao đối với tập dữ liệu cho trước nhưng khi áp dụng với tập dữ liệu bên ngoài thì độ chính xác giảm một cách đáng kể), một vài nơ-ron bị loại bỏ một cách ngẫu



Hình 2.9: Fully connected layer.

nhiên. Việc này giúp vecto đầu ra không bị phụ thuộc hoàn toàn vào giá trị của vecto đầu vào, kỹ thuật này được gọi là drop out.

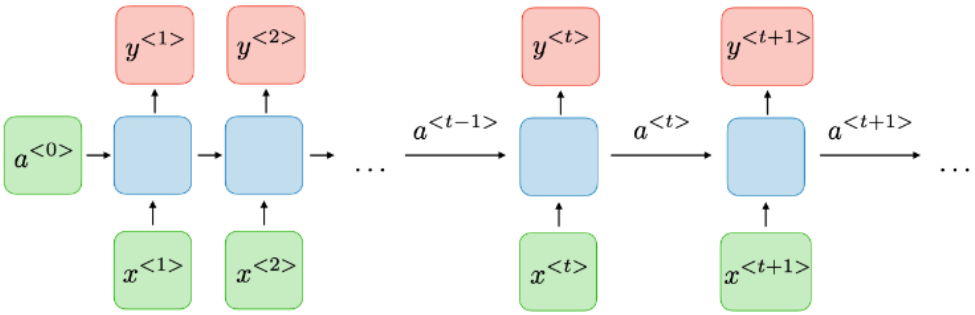


Hình 2.10: Drop out.

2.3 Mô hình RNN

Như đã biết, Neural Network gồm 3 phần: Input layer, Hidden layer và Output layer. Các đầu vào và đầu ra trong mạng này đều độc lập với nhau do đó mô hình này không phù hợp với những bài toán dạng chuỗi như mô tả, hoàn thành câu,... vì những dự đoán tiếp theo sẽ phụ thuộc vào vị trí trước của nó. RNN ra đời với ý tưởng chính là để giải quyết các bài toán đó.

RNN (Recurrent Neural Network) là một kiến trúc mạng nơ-ron có khả năng xử lý và phân tích các dữ liệu tuần tự. Trong RNN, mỗi nút (hoặc đơn vị) thực hiện các phép tính trên đầu vào hiện tại và trạng thái ẩn được truyền tiếp từ đơn vị trước đó.

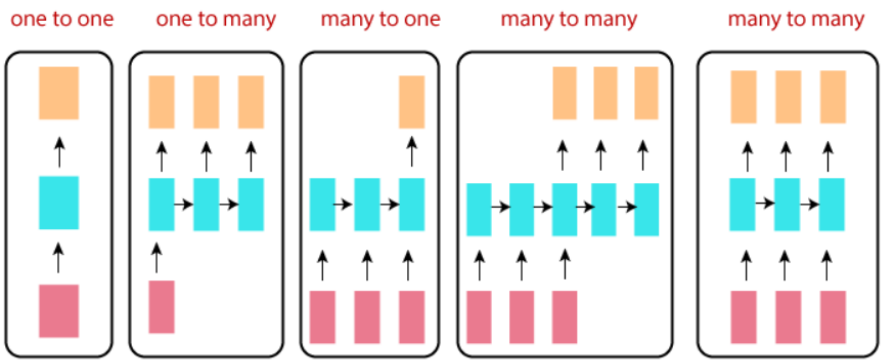


Hình 2.11: Recurrent neural network.

RNN được chia ra thành bốn loại:

- One to one: Đây còn được gọi là mạng Neural thuần túy. Nó xử lý một kích thước cố định của đầu vào với kích thước cố định của đầu ra, nơi chúng độc lập với thông tin đầu ra trước đó.
- One to many: Nó xử lý một kích thước cố định của thông tin làm đầu vào cung cấp một chuỗi dữ liệu làm đầu ra.
- Many to one: Bài toán có nhiều input nhưng chỉ có 1 output, ví dụ bài toán phân loại hành động trong video, input là nhiều ảnh (frame) tách ra từ video, ouptut là hành động trong video.

- Many to many: Bài toán có nhiều input và nhiều output, ví dụ bài toán dịch từ tiếng Anh sang tiếng Việt, input là 1 câu gồm nhiều chữ: “I love Vietnam” và output cũng là 1 câu gồm nhiều chữ “Tôi yêu Việt Nam”.



Hình 2.12: Các loại RNN.

Tuy nhiên mô hình RNN cũng tồn tại những hạn chế rõ ràng [6]:

- RNN có nhiệm vụ đưa thông tin đi kịp thời. Tuy nhiên, việc truyền tất cả các thông tin này là một việc khá khó khăn bởi thời gian quá dài. Điều đó dẫn đến RNN có thể gặp phải vanishing gradient (hệ số phản hồi không thể thay đổi khi các trạng thái mới đi qua mô hình). Việc này đồng nghĩa với việc mô hình không thể học được từ thông tin do đó kết quả đưa ra sẽ bị sai.
- Không thể mô hình hóa các thông tin dài hạn.
- RNN yêu cầu đầu vào mỗi chuỗi có độ dài bằng nhau do đó khó khăn trong việc xử lý các chuỗi có độ dài khác nhau.

2.4 Mô hình LSTM

Xuất phát từ những nhược điểm kể trên của RNN truyền thống, mô hình LSTM (Long Short – Term Memory) đã được ra đời. Dù mô hình LSTM có những kết nối phản hồi như mô hình RNN, nhưng quá trình xử lý không chỉ các điểm dữ liệu đơn lẻ mà là toàn bộ chuỗi dữ liệu. Từ đó kết quả trở lên lý tưởng hơn để xử lý và dự đoán dữ liệu, LSTM có nhiều ứng dụng trong nhận dạng chữ viết tay, nhận diện giọng nói, phát hiện giọng nói, trò chơi,...[7]

2.4.1 Short term và Long term memory

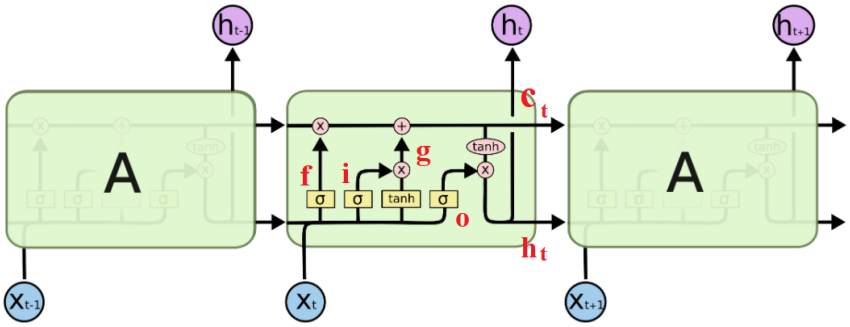
Short term memory là quá trình học từ những trạng thái được lưu trữ thông tin của layer trước tới layer sau khi thông tin đó được mang qua state nhất định. Nó là giá trị lưu trữ để tính toán đầu ra dự đoán đối với giá trị input. Short term memory của mô hình RNN sẽ tiếp tục liện tục qua nhiều state và khó kiểm soát và gặp vấn đề.

Long term memory là quá trình xác định thông tin đi xa hơn sau hàng ngàn state và sẽ được sử dụng khi cần. Giá trị long term memory sẽ được lưu trữ mới mỗi khi mộtshort term memory tính toán ra state tiếp theo của layer.

Để không xảy ra hiện tượng vanishing gradient, hàm kích hoạt được sử dụng là tanh và sigmoid. Giá trị đầu ra của hàm tanh bị giới hạn từ -1 < y < 1; đầu ra của hàm sigmoid giới hạn từ 0 < y < 1 sẽ điều chỉnh đầu ra như một hàm xác suất để dự đoán giống như các bài toán của Machine Learning. Kể cả đầu vào là một hình ảnh với ma trận hệ số màu thì các giá trị vẫn sẽ cùng giữ các đặc trưng (tỷ lệ chênh lệch) khi đi qua mô hình.

2.4.2 *Cấu trúc mô hình LSTM*

Cấu trúc của mô hình LSTM bao gồm một chuỗi các đơn vị lặp lại gọi là LSTM cells.



Hình 2.13: Mô hình LSTM.

Mỗi LSTM cell bao gồm 3 cổng chính [8]:

- Forget gate (f): Thực hiện việc quyết định thông tin nào từ trước đó sẽ được lưu lại trong bộ nhớ.
- Input gate (i): Thực hiện việc quyết định những thông tin mới nào sẽ được lưu lại trong bộ nhớ dài hạn.
- Output gate (o): Thực hiện xem thông tin nào được truyền ra ngoài để phục vụ cho mục đích dự đoán.

Ngoài ra thành phần của LSTM cell còn có cell state (c_t). Là bộ nhớ trong của LSTM, nó là tổng hợp của bộ nhớ trước c_{t-1} đã được lọc qua Forget gate, cộng với trạng thái ẩn g đã được lọc bởi Input gate sẽ mang thông tin nào quan trọng truyền đi xa hơn và sẽ được dùng khi cần. Đây chính là long term memory.

Cơ chế hoạt động đặc biệt và cổng (gate) của LSTM giúp mô hình có khả năng lưu trữ thông tin dài hạn và xử lý hiệu quả các chuỗi có độ dài khác nhau, giúp khắc phục được các nhược điểm của RNN. Tuy nhiên LSTM vẫn tồn tại điểm yếu là độ phức tạp của nó khiến thời gian xử lý lớn hơn nhiều so với mô hình RNN thông thường.

Chương 3 XÂY DỰNG MÔ HÌNH

3.1 Xây dựng mô hình

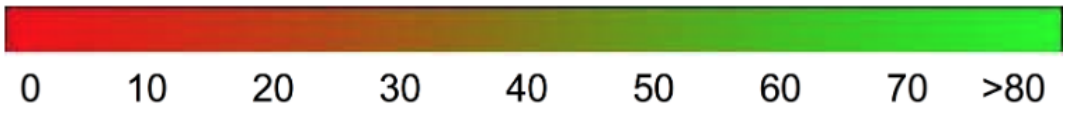
3.2 BLEU Score

BLEU (BiLingual Evaluation Understudy) là một phương pháp sử dụng trong dịch máy (Machine translation). BLEU đánh giá một câu thông qua việc so sánh khớp giữa câu máy sinh ra (candidate) và câu mẫu (preference), giá trị của đánh giá từ khoảng 0 tới 1. Giá trị 0 thể hiện việc đầu ra sai hoàn toàn và giá trị 1 thể hiện việc đúng hoàn toàn. Thông thường giá trị BLEU đạt trên 40% đã được coi là rất tốt [9].

BLEU đánh giá mô tả thông qua [10]:

- Adequacy: Mô tả đầy đủ thông tin trong ảnh
- Fluency: Đúng ngữ pháp, cấu trúc câu

BLEU Score	Interpretation
< 10	Almost useless
10 - 19	Hard to get the gist
20 - 29	The gist is clear, but has significant grammatical errors
30 - 40	Understandable to good translations
40 - 50	High quality translations
50 - 60	Very high quality, adequate, and fluent translations
> 60	Quality often better than human



Hình 3.1: Đánh giá BLEU.

$$\text{BLEU} = \underbrace{\min\Big(1, \exp\big(1 - \frac{\text{reference-length}}{\text{output-length}}\big)\Big)}_{\text{brevity penalty}} \underbrace{\Big(\prod_{i=1}^{4} precision_i\Big)^{1/4}}_{\text{n-gram overlap}}$$

with

$$precision_i = \frac{\sum_{\text{snt} \in \text{Cand-Corpus}} \sum_{i \in \text{snt}} \min(m^i_{cand}, m^i_{ref})}{w^i_t = \sum_{\text{snt}' \in \text{Cand-Corpus}} \sum_{i' \in \text{snt}'} m^{i'}_{cand}}$$

Hình 3.2: Tính toán đánh giá BLEU.

Đánh giá BLEU được tính toán như sau:

Công thức tính bao gồm hai phần [11]:

- Brevity penalty: Dùng để hạn chế việc đánh giá tốt cho các câu ngắn và kém cho các câu dài hơn.
- N-gram overlap: Hoạt động như một hàm tính độ chính xác của đầu ra.

trong đó

- m^i_{cand}: Số lượng i-gram trong candidate khớp với reference.
- m^i_{ref}: Số lượng i-gram trong reference.
- w^i_t: Tổng số i-gram trong candidate.

3.3 Nhận xét kết quả

Tài liệu tham khảo

[1] Volodymyr Kovenko. *Emotional attitude captioning dataset*. URL: https://data.mendeley.com/datasets/dym6p2pvbt/1.

[2] Shadab Hussain. *Flickr8 dataset*. URL: https://www.kaggle.com/datasets/shadabhussain/flickr8k.

[3] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, et al. "Convolutional neural networks: an overview and application in radiology". In: *Insights into imaging* 9 (2018), pp. 611–629.

[4] Manav Mandal. *Introduction to Convolutional Neural Networks (CNN)*. URL: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/#be44.

[5] Diego Unzueta. *Fully Connected Layer vs. Convolutional Layer: Explained*. URL: https://builtin.com/machine-learning/fully-connected-layer.

[6] Avijeet Biswal. *Disadvantages of Recurrent Neural Network*. URL: https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn?fbclid=IwAR2mwkQG8bpGKALoRSrkAukOxGJiULNnm2vT1-84ShvtNdRP

[7] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[8] Huyen Nguyen Thanh. *LSTM*. URL: https://viblo.asia/p/recurrent-neural-network-tu-rnn-den-lstm-gGJ597z1ZX2?fbclid=IwAR2CskZx40jmzFQ5Cts4WRd0X2M4bOBUG_EFC15NZO-W-IfRU9ZnBBlNr2M.

[9] Alon Lavie. "Evaluating the output of machine translation systems". In: *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas: Tutorials*. 2010.

[10] Kishore Papineni, Salim Roukos, Todd Ward, et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[11] Google Cloud. *Evaluating models*. URL: https://cloud.google.com/translate/automl/docs/evaluate#bleu.

Phụ lục A MÃ NGUỒN