ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

BÀI TẬP LỚN

CÁCH TIẾP CẬN HIỆN ĐẠI TRONG XỬ LÝ NGÔN NGỮ TỰ NHIÊN (055256)

ĐỀ TÀI:

SENTIMENT ANALYSIS

GVHD: Quản Thành Thơ
Lớp: 1
Sinh viên thực hiện: 1911314 – Lương Thị Quỳnh Hương

Tp. Hồ Chí Minh, Tháng 11/2022

Mục lục

A. Giới thiệu đề tài

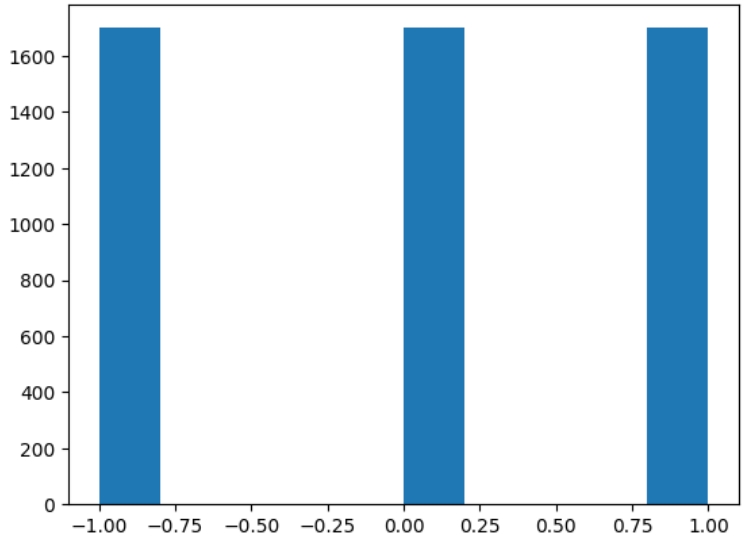
Sentiment analysis - phân tích quan điểm là một kỹ thuật xử lý ngôn ngữ tự nhiên để xác định dữ liệu là tích cực, tiêu cực hay trung lập. Phân tích quan điểm thường được biểu diễn bằng dữ liệu chữ giúp các doanh nghiệp giám sát quan điểm của khách hàng về thương hiệu hoặc sản phẩm và từ đó có thể hiểu được nhu cầu của khách hàng.

Nhờ sự phát triển không ngừng của học máy mà hiện nay ta có nhiều mô hình hiệu quả để giải quyết bài toán phân tích quan điểm. Trong bài tập lớn này, em sẽ hiện thực ba mô hình: CNN, LSTM và kết hợp CNN với LSTM đồng thời so sánh các mô hình này trong giải quyết bài toán phân tích quan điểm.

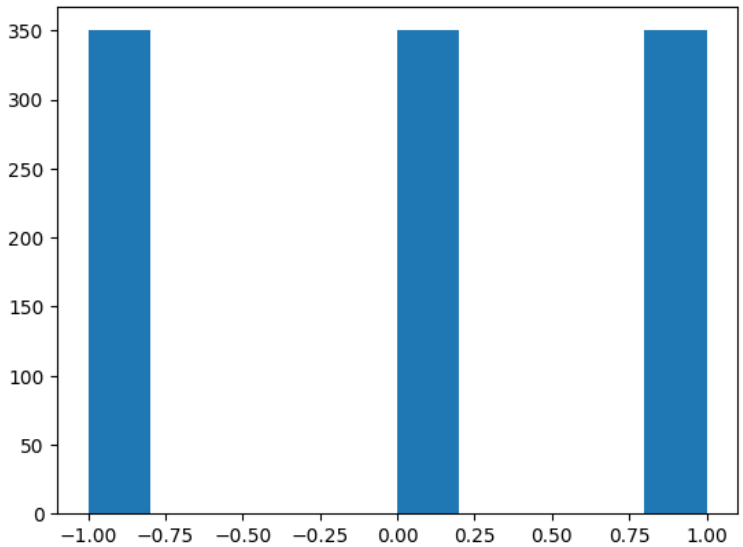
B. Tiền sử lý dữ liệu

I. Giới thiệu tập dữ liệu

Tập dữ liệu được dùng trong bài tập lớn này là vlsp_sentiment gồm một file dùng để train và một file dùng để test. Tập dữ liệu gồm hai cột là "Class" và cột "Data". Trong đó, cột "Class" có ba loại giá trị tương ứng với ba loại thái độ ứng với từng bình luận trong cột "Data": 1 - positive, 0 - neutral và (-1) - negative. Tổng các record ở tập train là 5100 và tập test là 1050. Dưới đây là biểu đồ trực quan hóa số lượng của các loại thái độ ở tập train và test:



Hình 1: *Số lượng của các loại thái độ ở tập train*



Hình 2: *Số lượng của các loại thái độ ở tập test*

Qua hai hình trên có thể thấy rằng số lượng mỗi loại thái độ ở các tập dữ liệu là bằng nhau, 1700 record cho mỗi loại thái độ ở tập train và 350 record cho mỗi thái độ ở tập test.

II. Tiền xử lý dữ liệu

Trước khi dùng dữ liệu ở tập train để train từng mô hình, em thực hiện tiền xử lý dữ liệu như sau:

- Mã hóa các nhãn ở cột "Class": (-1) thành $[1, 0, 0]$, 0 thành $[0, 1, 0]$ và 1 thành $[0, 0, 1]$
- Bỏ đi các ký tự là số ở cột "Data" vì chúng không mang nhiều ý nghĩ về mặt cảm xúc.
- Vì tập dữ liệu sử dụng tiếng Việt nên sử dụng ViTokenizer để tách từ sau đó vector hóa các từ.
- Tiến hành padding để đưa các record về cùng số chiều, đúng 300 chiều cho mỗi record. Khi đó ta có shape của X train và X validation tensor là $(5100, 300)$ và shape của label train và validation tensor là $(5100, 3)$.
- Dùng CBOW (file vi-model-CBOW.bin) để mã hóa các vector.

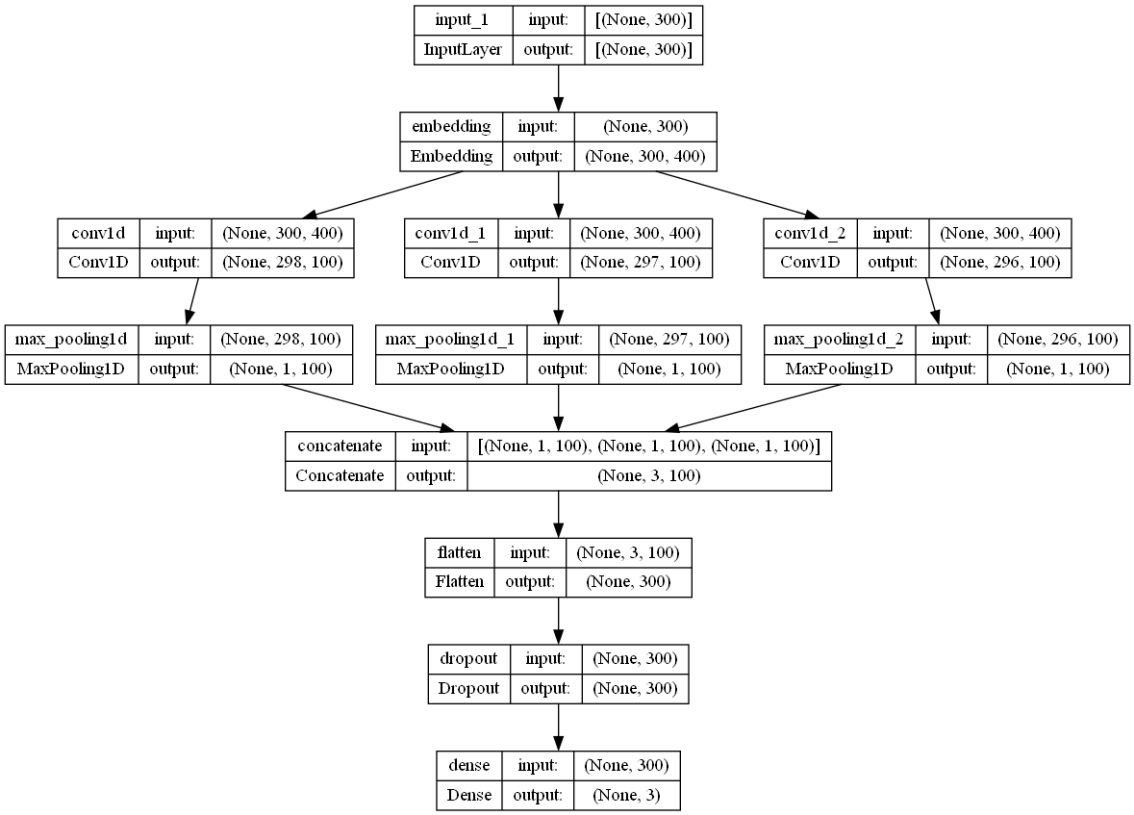
C. Sử dụng các mô hình để giải quyết bài toán sentiment analysis

Tất cả source code trong bài tập lớn này nằm ở link github sau:
https://github.com/LuongThiQuynhHuong/NLP_assignment_221_1911314.git

Để huấn luyện cho các mô hình sau đây, tập train sẽ được chia ra hai phần, 80% để train và 20% để validate và vì tập dữ liệu đang sắp xếp theo từng nhóm nhãn nên phải xáo trộn (shuffle) tập huấn luyện.

I. Mô hình Convolutional Neural Network - CNN

Kiến trúc mô hình CNN đã dùng được trực quan hóa trong sơ đồ sau đây:



Hình 3: *Kiến trúc mô hình CNN*

Mô hình CNN này được thiết lập như sau:

- Sử dụng 3 loại lớp tích chập với kích thước filter lần lượt là 3, 4, 5 để học được mối liên quan giữa 3, 4, 5 từ đứng gần nhau, và số lượng filter đầu ra trong lớp tích chập là 100.
- Hàm kích hoạt của lớp tích chập là hàm relu.
- Sử dụng kỹ thuật Dropout với xác xuất 0.5 để tránh overfitting.
- Hàm kích hoạt ở tầng output là softmax.

- Tính loss bằng categorical_crossentropy và dùng thuật toán tối ưu là adam.

Lịch sử train mô hình trong 27 epoch và batch size bằng 256 như sau:

```
Epoch 1/32
16/16 [==============================] - 17s 1s/step - loss: 7.4024 - accuracy: 0.4515 - val_loss: 7.7539 - val_accuracy: 0.0402
Epoch 2/32
16/16 [==============================] - 17s 1s/step - loss: 5.4219 - accuracy: 0.6208 - val_loss: 6.7011 - val_accuracy: 0.0833
Epoch 3/32
16/16 [==============================] - 17s 1s/step - loss: 4.5098 - accuracy: 0.7199 - val_loss: 6.1746 - val_accuracy: 0.0794
Epoch 4/32
16/16 [==============================] - 19s 1s/step - loss: 3.8664 - accuracy: 0.7980 - val_loss: 5.7598 - val_accuracy: 0.0490
Epoch 5/32
16/16 [==============================] - 18s 1s/step - loss: 3.3514 - accuracy: 0.8441 - val_loss: 5.1453 - val_accuracy: 0.0804
Epoch 6/32
16/16 [==============================] - 19s 1s/step - loss: 2.9106 - accuracy: 0.8824 - val_loss: 4.4391 - val_accuracy: 0.1461
Epoch 7/32
16/16 [==============================] - 18s 1s/step - loss: 2.5481 - accuracy: 0.9015 - val_loss: 4.6065 - val_accuracy: 0.0696
Epoch 8/32
16/16 [==============================] - 18s 1s/step - loss: 2.2242 - accuracy: 0.9257 - val_loss: 3.9309 - val_accuracy: 0.1255
Epoch 9/32
16/16 [==============================] - 17s 1s/step - loss: 1.9625 - accuracy: 0.9326 - val_loss: 4.3771 - val_accuracy: 0.0363
Epoch 10/32
16/16 [==============================] - 18s 1s/step - loss: 1.7365 - accuracy: 0.9348 - val_loss: 3.5007 - val_accuracy: 0.1225
Epoch 11/32
16/16 [==============================] - 19s 1s/step - loss: 1.5354 - accuracy: 0.9449 - val_loss: 3.4575 - val_accuracy: 0.0941
Epoch 12/32
16/16 [==============================] - 18s 1s/step - loss: 1.3686 - accuracy: 0.9502 - val_loss: 3.4734 - val_accuracy: 0.0755
Epoch 13/32
16/16 [==============================] - 18s 1s/step - loss: 1.2261 - accuracy: 0.9527 - val_loss: 3.4371 - val_accuracy: 0.0510
Epoch 14/32
16/16 [==============================] - 18s 1s/step - loss: 1.0972 - accuracy: 0.9610 - val_loss: 2.9837 - val_accuracy: 0.1216
Epoch 15/32
16/16 [==============================] - 17s 1s/step - loss: 0.9973 - accuracy: 0.9598 - val_loss: 3.3842 - val_accuracy: 0.0520
Epoch 16/32
16/16 [==============================] - 17s 1s/step - loss: 0.9113 - accuracy: 0.9596 - val_loss: 3.1051 - val_accuracy: 0.0657
Epoch 17/32
16/16 [==============================] - 19s 1s/step - loss: 0.8355 - accuracy: 0.9623 - val_loss: 2.6551 - val_accuracy: 0.1363
Epoch 18/32
16/16 [==============================] - 19s 1s/step - loss: 0.7807 - accuracy: 0.9566 - val_loss: 2.5160 - val_accuracy: 0.1833
Epoch 19/32
16/16 [==============================] - 20s 1s/step - loss: 0.7254 - accuracy: 0.9625 - val_loss: 2.8936 - val_accuracy: 0.0882
Epoch 20/32
16/16 [==============================] - 20s 1s/step - loss: 0.6805 - accuracy: 0.9600 - val_loss: 2.8029 - val_accuracy: 0.0912
Epoch 21/32
16/16 [==============================] - 19s 1s/step - loss: 0.6410 - accuracy: 0.9664 - val_loss: 2.4199 - val_accuracy: 0.1637
Epoch 22/32
16/16 [==============================] - 20s 1s/step - loss: 0.6030 - accuracy: 0.9664 - val_loss: 2.4318 - val_accuracy: 0.1529
Epoch 23/32
16/16 [==============================] - 20s 1s/step - loss: 0.5853 - accuracy: 0.9676 - val_loss: 2.1251 - val_accuracy: 0.2343
Epoch 24/32
16/16 [==============================] - 18s 1s/step - loss: 0.5771 - accuracy: 0.9588 - val_loss: 2.4703 - val_accuracy: 0.1480
Epoch 25/32
16/16 [==============================] - 20s 1s/step - loss: 0.5450 - accuracy: 0.9650 - val_loss: 2.5515 - val_accuracy: 0.1216
Epoch 26/32
16/16 [==============================] - 18s 1s/step - loss: 0.5278 - accuracy: 0.9650 - val_loss: 2.6888 - val_accuracy: 0.1069
Epoch 27/32
16/16 [==============================] - 18s 1s/step - loss: 0.5011 - accuracy: 0.9706 - val_loss: 2.9580 - val_accuracy: 0.0608
Epoch 27: early stopping
```

Hình 4: *CNN history*

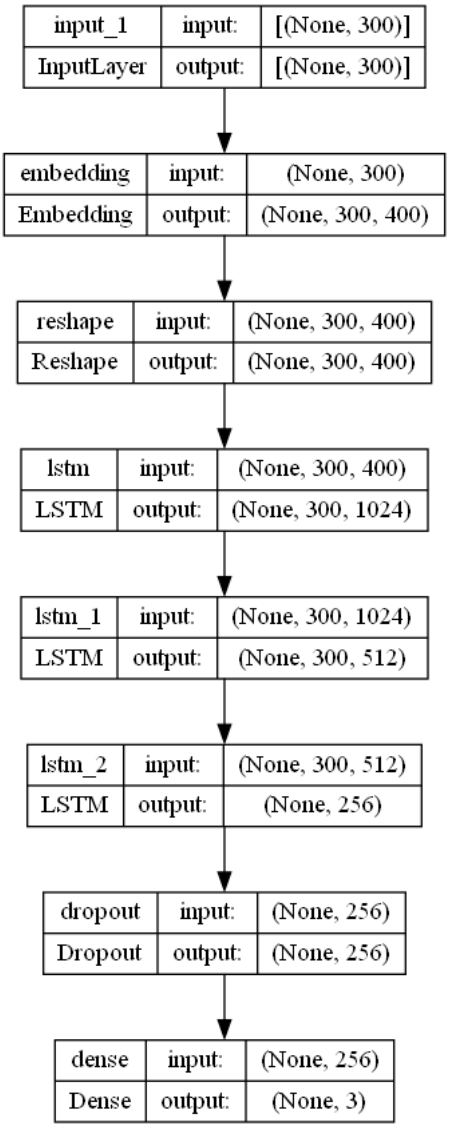
Kết quả của mô hình CNN được tổng hợp như trong bảng sau:

	Loss	**Accuracy**
Train	0.5011	97.06%
Validate	2.9580	6.08%
Test	1.34	61.9%

Qua bảng trên ta có thể thấy có sự overfit trên tập train của tập dữ liệu, là độ chính xác trên tập validate rất thấp. Điều này xảy ra có thể vì tập dữ liệu vẫn còn khá ít.

II. Mô hình Long Short Term Memory - LSTM

Kiến trúc mô hình LSTM đã dùng được trực quan hóa trong sơ đồ sau đây:



Hình 5: *Kiến trúc mô hình LSTM*

Mô hình LSTM này được thiết lập như sau:

- Dữ liệu ban đầu lần lượt đi qua ba lớp LSTM với số node lần lượt là 1024, 512 và 256.
- Hàm kích hoạt của lớp LSTM là hàm tanh.
- Sử dụng kỹ thuật Dropout với xác xuất 0.5 để tránh overfitting.
- Hàm kích hoạt ở tầng output là softmax.
- Tính loss bằng categorical_crossentropy và dùng thuật toán tối ưu là adam.

Lịch sử train mô hình trong 7 epoch và batch size bằng 256 như sau:

```
Epoch 1/10
16/16 [==============================] - 748s 48s/step - loss: 1.2188 - accuracy: 0.4355 - val_loss: 1.8516 - val_accuracy: 0.0029
Epoch 2/10
16/16 [==============================] - 919s 59s/step - loss: 0.9203 - accuracy: 0.6074 - val_loss: 1.6342 - val_accuracy: 0.1529
Epoch 3/10
16/16 [==============================] - 1176s 75s/step - loss: 0.8125 - accuracy: 0.6699 - val_loss: 1.5441 - val_accuracy: 0.1431
Epoch 4/10
16/16 [==============================] - 1335s 84s/step - loss: 0.6752 - accuracy: 0.7569 - val_loss: 2.3673 - val_accuracy: 0.1078
Epoch 5/10
16/16 [==============================] - 1366s 86s/step - loss: 0.5581 - accuracy: 0.8086 - val_loss: 1.5679 - val_accuracy: 0.3735
Epoch 6/10
16/16 [==============================] - 1475s 93s/step - loss: 0.4382 - accuracy: 0.8578 - val_loss: 2.4509 - val_accuracy: 0.1941
Epoch 7/10
16/16 [==============================] - 1494s 94s/step - loss: 0.3472 - accuracy: 0.8897 - val_loss: 3.0812 - val_accuracy: 0.1157
Epoch 7: early stopping
```

Hình 6: *LSTM history*

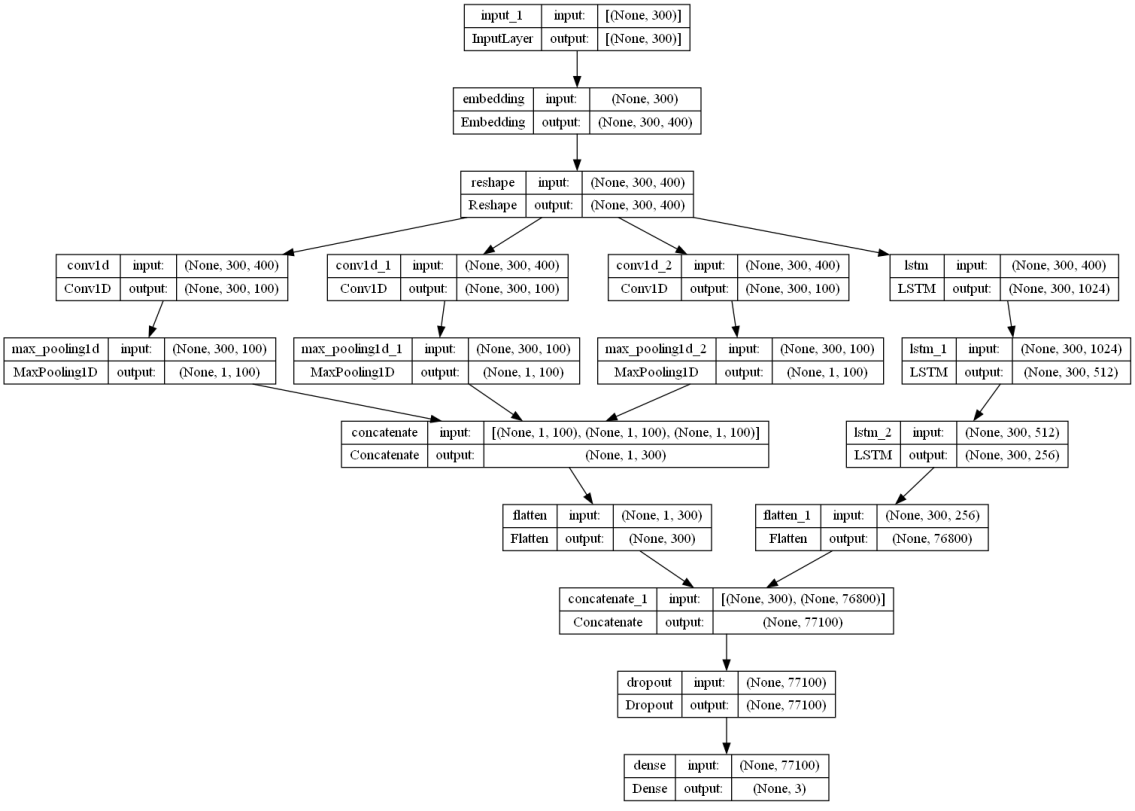
Kết quả của mô hình LSTM được tổng hợp như trong bảng sau:

	Loss	**Accuracy**
Train	0.3472	88.97%
Validate	3.0812	11.57%
Test	1.21	61.9%

Qua bảng trên ta có thể thấy có sự overfit trên tập train của tập dữ liệu, là độ chính xác trên tập validate rất thấp. Điều này xảy ra có thể vì tập dữ liệu vẫn còn khá ít.

III. Kết hợp mô hình CNN và LSTM

Kiến trúc mô hình kết hợp giữa mô hình CNN và LSTM đã dùng được trực quan hóa trong sơ đồ sau đây:



Hình 7: *Kiến trúc mô hình kết hợp CNN và LSTM*

Mô hình kết hợp CNN và LSTM này được thiết lập như sau:

- Dữ liệu ban đầu lần lượt đi qua ba loại lớp tích chập với kích thước filter lần lượt là 3, 4, 5 và một lớp LSTM với số node là 1024, output từ lớp LSTM này sẽ đi qua thêm hai lớp LSTM khác có số node lần lượt là 512 và 256.
- Hàm kích hoạt của lớp LSTM là hàm tanh và hàm kích hoạt của lớp tích chập là hàm relu.
- Sử dụng kỹ thuật Dropout với xác xuất 0.5 để tránh overfitting.
- Hàm kích hoạt ở tầng output là softmax.
- Tính loss bằng categorical_crossentropy và dùng thuật toán tối ưu là adam.

Lịch sử train mô hình trong 7 epoch và batch size bằng 256 như sau:

```
Epoch 1/10
16/16 [==============================] - 652s 42s/step - loss: 4.6906 - accuracy: 0.4534 - val_loss: 4.2045 - val_accuracy: 0.0049
Epoch 2/10
16/16 [==============================] - 863s 56s/step - loss: 2.5100 - accuracy: 0.5983 - val_loss: 2.8700 - val_accuracy: 0.0000e+0
Epoch 3/10
16/16 [==============================] - 1167s 75s/step - loss: 1.4365 - accuracy: 0.6703 - val_loss: 1.6970 - val_accuracy: 0.2510
Epoch 4/10
16/16 [==============================] - 1393s 89s/step - loss: 0.9462 - accuracy: 0.7419 - val_loss: 1.8397 - val_accuracy: 0.2294
Epoch 5/10
16/16 [==============================] - 1501s 94s/step - loss: 0.6954 - accuracy: 0.7926 - val_loss: 1.9587 - val_accuracy: 0.2814
Epoch 6/10
16/16 [==============================] - 1692s 107s/step - loss: 0.5375 - accuracy: 0.8350 - val_loss: 2.9786 - val_accuracy: 0.1422
Epoch 7/10
16/16 [==============================] - 1725s 108s/step - loss: 0.3949 - accuracy: 0.8890 - val_loss: 3.9031 - val_accuracy: 0.0549
Epoch 7: early stopping
```

Hình 8: *CNN and LSTM history*

Kết quả của mô hình CNN kết hợp với LSTM được tổng hợp như trong bảng sau:

	Loss	**Accuracy**
Train	0.3949	88.9%
Validate	3.9031	5.49%
Test	1.37	60.67%

Qua bảng trên ta có thể thấy có sự overfit trên tập train của tập dữ liệu, là độ chính xác trên tập validate rất thấp. Điều này xảy ra có thể vì tập dữ liệu vẫn còn khá ít.

D. Tổng kết

I. Tổng hợp kết quả

Các kết quả từ các mô hình ở phần trên được tổng hợp trong bảng sau:

	Loss	Accuracy
CNN_train	0.5011	97.06%
LSTM_train	0.3472	88.97%
CNN_and_LSTM_train	0.3949	88.9%
CNN_validate	2.958	6.08%
LSTM_validate	3.0812	11.57%
CNN_and_LSTM_validate	3.9031	5.49%
CNN_test	1.34	61.9%
LSTM_test	1.21	61.9%
CNN_and_LSTM_test	1.37	60.67%

**Nhận xét:* Sau khi áp dụng các mô hình trên để giải quyết bài toán sentiment analysis trên tập dữ liệu đã giới thiệu ở đầu báo cáo, em có một số nhận xét như sau:

- Cả ba mô hình đều có độ chính xác trên tập train rất cao. Mô hình CNN có accuracy cao nhất và gần bằng 100%. Mô hình LSTM và mô hình kết hợp cNN và LSTM có accuracy gần bằng 90%.
- Độ chính xác trên tập validate của cả ba mô hình đều rất thấp. Accuracy cao nhất là của mô hình LSTM nhưng cũng chỉ gần 12%.
- Độ chính xác của cả ba mô hình trên tập test nhìn chung khá giống nhau, đều khoảng 61 - 62%.

II. Kết luận

Sau khi hoàn thành thí nghiệm, em có một số kết luận như sau:

- Có dấu hiệu overfitting ở tập train ở cả ba mô hình đã thí nghiệm. Điều này xảy ra có thể là do bộ dữ liệu vẫn còn ít record.
- Độ chính xác trên tập validate của cả ba mô hình đều rất thấp. Điều này xảy ra có thể là do lượng dữ liệu dùng để validate quá ít.
- Độ chính xác ở cả ba mô hình trên nhìn chung khá tương tự nhau và không cao, không có mô hình nào tốt hơn hẳn cả. Đây có thể là do tập dữ liệu chưa đủ lớn, hoặc các thông số truyền vào các mô hình vẫn còn chưa được hợp lý hoặc là bước tiền xử lý dữ liệu vẫn chưa được tối ưu.
- Trong quá trình huấn luyện, thời gian huấn luyện mô hình CNN nhanh hơn rất nhiều so với hai mô hình còn lại.

A Tài liệu tham khảo

Tài liệu

[1] https://www.analyticsvidhya.com/blog/2021/06/natural-language-processing-sentiment-analysis-using-lstm/

[2] Quản Thành Thơ, *MẠNG NƠ-RON NHÂN TẠO: TỪ HỒI QUY ĐẾN HỌC SÂU*, NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA (2021)

[3] https://machinelearningmastery.com/develop-word-embedding-model-predicting-mov

[4] https://medium.ninja/@mrunal68/text-sentiments-classification-with-cnn-and-lst