BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**BÁO CÁO TIẾN ĐỘ**

**Đề tài**

**PHÂN TÍCH CẢM XÚC**

**SỬ DỤNG MÔ HÌNH LSTM**

**Sinh viên: Nguyễn Đức Tín**

**Mã số: B1801994**

**Khóa: 44**

**Cần Thơ, 10/2021**

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO TIẾN ĐỘ**

**Đề tài**

**PHÂN TÍCH CẢM XÚC**

**SỬ DỤNG MÔ HÌNH LSTM**

**Người hướng dẫn Sinh viên thực hiện**

**TS/Ths Lâm Nhựt Khang Nguyễn Đức Tín**

**Mã số: B1809194**

**Khóa: 44**

***Cần Thơ, 10/2021***

**LỤC MỤC**

**HÌNH ẢNH**

**PHỤ LỤC**

# CHƯƠNG I: GIỚI THIỆU

1. **Đăt vấn đề**

Ngày nay với sự phát triển phổ biến của Internet với các trang mạng xã hội như Facebook, Instagram, Twitter, … Các trang thương mại điện tử như Shope, Lazada, … Internet không chỉ còn là nơi cung cấp thông tin, quản bá dịch vụ, sản phẩm mà còn là nơi người dùng bày tỏ cảm xúc, ý kiến cá nhân, kinh nghiệm về các vấn đề thực tế mà chính họ đã trải qua thông qua các bình luận, nhận xét trên Internet. Việc phân tích những bình luận này sẽ xác định chính xác cảm xúc của người dùng để đưa ra kết luận một sản phẩm, dịch vụ nào đó là tốt hay không? Có thể triển khai rộng rãi hay không? [1]

Việc phân loại bình luận theo các loại cảm xúc khác nhau là một điều cần thiết tuy nhiên trong thực tế việc phân loại này sẽ rất khó khăn với lượng dữ liệu cần phân tích quá lớn vì vậy để giải quyết vấn đề này em đề nghị xây dựng một mô hình máy học có giám sát giúp phân loại cảm xúc của bình luận dựa trên các nhãn đã được định nghĩa trước.

1. **Mục tiêu của đề tài**

Mục tiêu của đề tài là nghiên cứu các phương pháp để số hóa dữ liệu từ các từ thành các vectơ tương ứng và sử dụng mạng Long Short Term Memory (LSTM) để tiến hành xây dựng mô hình phân loại cảm xúc.

1. **Đối tượng phạm vi nghiên cứu**

* Các phương pháp giúp mô phỏng một từ trong không gian vectơ.
* Tìm hiểu lý thuyết mạng Long Short Term Memory (LSTM) và áp dụng để xây dựng mô hình phân loại cảm xúc
* Tập dữ liệu sử dụng trong đề tài này là các bình luận được gán nhãn tại trang web [foody.vn](file:///C:\Users\Admin\AppData\Roaming\Microsoft\Word\foody.vn) đối với tiếng việt và một số câu nhận xét bằng tiếng anh đã gán nhãn

1. **Phương pháp nghiên cứu**

Phương pháp nghiên cứu tài liệu: đọc các tài liệu có liên quan đến đề tài từ sách chuyên ngành, các bài báo khoa học, các trang web có liên quan đến vấn đề đang nghiên cứu, nghiên cứu các kĩ thuật lập trình trong Python, các sử dụng Google Colab để thực nghiệm mô hình.

Phương pháp nghiên cứu thực nghiệm:

* + Tiền sử lý dữ liệu và xây dựng mô hình bằng ngôn ngữ Python.
  + Tiến hành thực nghiệm trên tập dữ liệu.
  + So sánh và đánh giá trên kết quả thu được từ tập dữ liệu thực nghiệm

1. **Nội dung nghiên cứu**
2. **Bố cục của niên luận**

Nội dung niên luận được chia thành 3 chương. Giới thiệu vấn đề, mục tiêu đề tài, đối tượng và phạm vi nghiên cứu, phương pháp nghiên cứu được trình bày ở chương 1. Cơ sở lý thuyết sẽ được trình bày ở chương 2. Chương 3 : Cài đặt mô hình và thực nghiệm. Kết luận và hướng phát triển sẽ được trình bày tại chương 4.

## CHƯƠNG II CƠ SỞ LÝ THUYẾT

1. **Giới thiệu về máy học**

Máy học (Machine Learning) về cơ bản là khả năng xây dựng hệ thống tự học hỏi dựa trên dữ liệu được cho trước [2]. Các cách phổ biến dùng để phân loại các thuật toán máy học là dựa trên phương thức học và dựa trên sự tương đồng về chức năng.

Dựa trên phương thức học các thuật toán máy học thường được chia thành 4 nhóm: Học có giám sát (Supervised Learning), Học không giám sát (Unsupervised Learning), Học bán giám sát (hay học kết hợp - Semi-supervised Learning) và Học tăng cường (Reinforcement Learning).

Dựa trên chức năng bao gồm một số nhóm thuật toán như: thuật toán hồi quy, huật toán dựa trên mẫu, Thuật toán chuẩn hoá, Thuật toán cây quyết định,…Nhưng phổ biến nhất là 2 loại chính là học có giám sát, và học không giám sát

* 1. **Học có giám sát**

Học có giám sát (Supervised Learning) là thuật toán dự đoán đầu ra (nhãn) của một dữ liệu mới dựa trên tập dữ liệu huấn luyện đã được gán nhãn thông qua việc huấn luyện một mô hình. Trong quá trình huấn luyện này mô hình mỗi khi dự đoán sai sẽ được tinh chỉnh tự động để có thể đạt được độ chính xác nhất định. Các thuật toán học giám sát này còn được phân ra làm 2 loại chính là Phân lớp (Classification) và Hồi quy (Regression). Đây là phương pháp phổ biến nhất trong máy học và cũng là phương pháp em sử dụng để thiết lập mô hình.

* 1. **Học không giám sát**

Trong thuật toán này ta không thể biết được chính xác đầu ra (nhãn) vì dùng dữ liệu huấn luyện chưa được gán nhãn. Vì vậy thuật toán này hướng đến việc tìm ra cấu trúc hoặc sự liên kết của dữ liệu để làm một việc gì đó như phân nhóm dựa trên những đặc điểm tương đồng hoặc giảm chiều của dữ liệu.

1. **Tách từ trong văn bản tiếng việt**

Khác với từ tiếng anh phân tách nhau bằng dấu cách, trong tiếng Việt dấu cách dùng để phân tách âm tiết chứ không phải từ [3]. Trong tiếng Việt bào gồm từ đơn và từ ghép và một từ hoàn toàn có thể thay đổi ý nghĩa khi đi cùng một từ khác nên việc tách từ và phân loại hết sức khó khăn.

Trong việc sử lý dữ liệu văn bản thành dữ liệu đầu vào cho một mô hình máy học như bài toán phân loại văn bản, tóm tắt văn bản, trích xuất thông tin quan điểm… Các văn bản hay cụ thể hơn là các câu sẽ được tách thành các từ và các từ này sẽ được số hóa thành các vector tương ứng để có thể làm dữ liệu đầu vào cho việc huấn luyện mô hình máy học.Có hai hướng tiếp cận chính là tiếp cận dựa trên “từ” và dựa trên “kí tự” trong bài viết này em chỉ đề cập đến hướng tiếp cận dựa trên “từ”. Các phương pháp phổ biến dựa trên đặc điểm của “từ” để tách từ trong văn bản Tiếng Việt có thể chia làm 3 hướng chính là: dựa trên thống kê (statistics - based), dựa trên từ điển ( dictionary – based) và tiếp cận lai (hydrid).

1. **Phương pháp tách từ dựa trên từ điển**

Đây là một phương pháp không quá phức tạp về cơ bản ta chỉ cần sử dụng một từ điển sẵn có để tiến hành so khớp các từ trong văn bản và các từ trong từ điển với nhau. Tùy vào cách so khớp mà ta có các phương pháp cụ thể khác nhau: so khớp từ dài nhất (longest matching), so khớp từ ngắn nhất (short matching), so khớp chồng lắp (overlap matching) và so khớp cực đại (maximum matching). Tuy nhiên độ hiệu quả của phương pháp này lại phụ thuộc hoàn toàn vào bộ từ điển được xây dựng nên trong trường hợp các từ cần sử lý quá đặc biệt mà không nằm trong từ điển thì độ hiệu quả trở nên rất thấp nhưng bù lại thời gian cho việc nghiên cứu và xử lý tách từ rất nhanh do không cần tiến hành huấn luyện. [4]

1. **Phương pháp tách từ dựa trên thống kê**

Với cách tiếp cận này các giải pháp cho việc tách từ thường dựa trên mô hình ngôn ngữ (language model – LM). Một mô hình ngôn ngữ đưuọc xây dựng bằng cách thống kê số lần xuất hiện hoặc đồng xuất hiện của các từ trong tập hợp các văn bản. Một số mô hình được sử dụng có thể kể đến như: mô hình n-grams, mô hình Markov ẩn (Hidden Markov Model), mô hình kết hợp trường xác suất có điều kiện và độ hỗn loạn cực đại, mô hình Pointwise,… [4]

1. **Phương pháp tiếp cận lai**

Có thể thấy cả hai phương pháp tiếp cận dựa trên từ điển và dựa trên thống kê đều có ưu nhược điểm của nó vì vậy để tận dụng tốt nhất các ưu điểm của mỗi loại tiếp cận phương pháp tiếp cận lai được đề suất. Một số phương pháp được đề suất của việc kết hợp này có thể kể đến như: kết hợp giữa mô hình ngôn ngữ Weighted Finite State Transducer (WFST) và mạng Neural , kết hợp mô hình so khớp cực đại và ngôn ngữ mô hình ngrams, hệ thống tách từ tiếng Việt WS4VN kết hợp giữa phƣơng pháp so khớp cực đại và mô hình Markov ẩn,…

1. **Biểu diễn từ thành vector số học**

Các từ sau khi được tách không thể trực tiếp đưa vào làm dữ liệu đầu vào để máy tính hiểu được mà ta cần chuyển chúng về dạng mà máy tính có thể hiểu được .Phương pháp tiêu chuẩn để biểu diễn đó là biểu diễn văn bản theo vector. Vì vậy ta cần phải biểu diễn các từ thành các vector số học tương ứng.

Hai cách chính để vector hóa văn bản bao gồm: Word Embedding cổ điển (nhúng từ) và Neural Embedding (vector hóa văn bản bằng mạng mạng nơ-ron) [5].

1. **Phương pháp Word Embedding cổ điển**
2. **Bag of Words (BoW)**

Đây là cách biểu diễn truyền thống phổ biến nhất.Phương pháp này cần có một tập hợp các từ cho trước được gọi là túi đựng từ. Ta sẽ biểu diễn mỗi từ thông qua một vector one-hot với số chiều bằng đúng số từ trong túi đựng. Tại vị trí tương ứng với vị trí của từ đó trong bộ từ điển sẽ được đánh dấu là 1, các vị trí còn lại sẽ được đánh dấu là 0.Tuy nhiên phương pháp này không thể hiển mối liên kết giữa các từ cũng như ý nghĩa của chúng. [5]

1. **TF-IDF (term frequency–inverse document frequency)**

Đây là phương pháp sử dụng trọng số của một từ thu được bằng việc thống kê thể hiện mức độ quan trọng của từ này trong toàn bộ dữ liệu đầu vào [6]

*Tf- term frequency*:tần xuất suất hiện của từ trong văn bản. Mỗi văn bản sẽ có tần xuất cảu từ khác nhau được tính dựa vào công thức:  (2.1)

Trong đó:

**i**:1..D

**:** Tần số suất hiện của từ trong văn bản

**:** Tổng số từ trong văn bản

*IDF- Inverse Document Frequency:* ước lượng mức độ quan trọng của từ. Do có một số từ xuất hiện rất nhiều nhưng không quan trọng trong thể hiện ý nghĩa của văn bản vì vậy cần phải giảm mức độ quan trọng xuống bằng cách sử dụng công thức sau:

(2.2)

Trong đó:

**:** Tổng số văn bản trong tập mẫu

**:** Số văn bản có suất hiện của từ

Vậy mức độ quan trọng cuối cùng của từ được xác định bằng công thứcTF- IDF như sau:

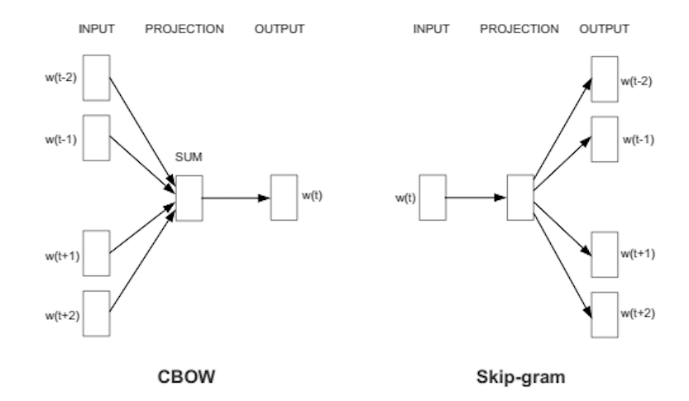
(2.3)

Ta có thể thấy rằng TF-IDF tỷ lệ thuận với tần số xuất hiện của từ và số văn bản xuất hiện từ đó nên phương pháp này có tính phân loại cao hơn tuy nhiên vẫn không thể hiện được ý nghĩa của từ.

1. **Phương pháp Neural Embedding**
2. **Word2Vec**

Phương pháp word to vector sẽ biểu diễn các từ có chung ngữ cảnh sẽ đặt gần nhau trong không gian vector gồm hai mô hình:

* Continuous skip-gram: dùng một từ để dự đoán các từ xung quanh. Trong mô hình này đầu vào là một one-hot vector của chỉ một từ duy nhất và mô hình sẽ cố gắng dự đoán các từ lân cận của từ này.
* CBOW (Continuous Bag of Word): dự đoán một từ dựa vào các từ lân cận. Hoàn toàn ngược lại với mô hình trên phương pháp này sẽ dùng các từ ở tầng đầu vào để dự đoán một mục tiêu gần nhất.



Hình 2.1 [5]

Mỗi phương pháp đều có ưu nhược điểm riêng. Skip-gram hiệu quả hơn với dữ liệu dào tạo nhỏ và với các cụm từ hiếm. CBOW cho thời gian huấn luyện ngắn hơn và hiệu quả với các từ thông dụng.

1. **GloVe**

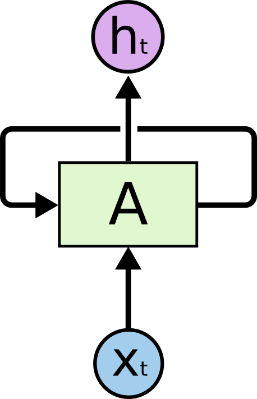
Với Word2vec cả hai phương pháp chỉ xem xét ngữ cảnh xung quanh mục tiêu mà không thể xét đến toàn văn bản. Thuật toán GloVe sử dụng ma trận đồng xuất hiện sử dụng trong thuật toán Distributional Embedding (một phương pháp khác trong Word Embedding cổ điển) kết hợp với phương pháp Neural Embedding làm cho vector có ý nghĩa và tỷ trọng hơn. [5]

1. **Mô hình SLTM**
2. **Giới thiệu về mạng hồi quy RNN**

Một bình luận hay một văn bản không thể chỉ gồm một hay một vài từ mà phải bao gồm nhiều từ và nhiều câu khác nhau. Do đó việc phân loại cảm xúc của bình luận không thể chỉ dựa vào một từ mà phải dựa trên ngữ cảnh của toàn bộ các từ đã được đọc trước đó. Điều này hoàn toàn dễ hiểu như việc bản thân ta không chỉ sử lý và hiểu một vấn đề dựa trên bối cảnh hiện tại mà ta còn xét đến những vấn đề xảy ra trước đó. Giống như việc ta đọc một văn bản, ta sẽ đọc từng từ và hiểu nó không chỉ dựa vào ý nghĩa từ này mà còn căn cứ cả các từ trước đó mà ta đã đọc. Ta không hề bỏ qua các từ trước đó để bắt đầu suy diễn từ đầu mà sự suy diễn này phải được duy trì liên tục.

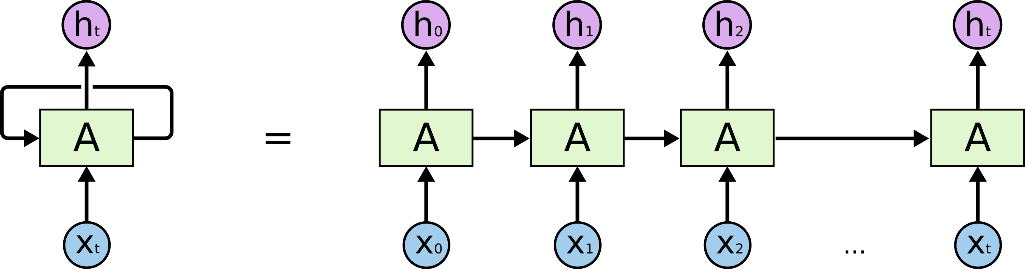
Các mạng nơ-ron truyền thống không thể thực hiện được điều này do không có sự kết nối giữa đầu vào và đầu ra với nhau. Mô hình này trở nên không phù hợp với nhiều bài toán như tự động sinh văn bản, dịch máy... Giả sử ta muốn dự đoán xác suất của một từ mới tiếp theo có thể suất hiện trong một câu thì ta không thể chỉ dựa vào từ hiện tại được mà còn cả các từ trước đó. Một ví dụ khác là việc dịch ngôn ngữ, công việc không đơn giản là ánh xạ một từ đến nghĩa của từ đó trong từ điển vì mỗi từ sẽ thay đổi ý nghĩa khi đi cùng một từ trước đó và một từ sẽ mang rất nhiều nghĩa trong từng bối cảnh khác nhau.

Mạng hồi quy (Recurrent Neural Networks) [7] hay viết tắt là mạng RNN giúp giải quyết các vấn đề đó. Nó chứa các vòng lặp bên trong và cho phép thông tin có thể lưu lại được. Mô hình được mô tả cơ bản như sau:



Hình 2.1 Mô hình mạng RNN [7]

Mô hình bao gồm một mạng nơ-ron hồi quy A với đầu vào là Xt và đầu ra là ht và có một vòng lặp làm cho thông tin truyền từ bước này sang bước khác của mạng nói cách khác là mỗi trạng thái mạng của mô hình là giống nhau nên ta có thể thu gọn thành một vòng lặp mà thôi nên bản chất nó vấn khá giống với mạng nơ-ron thông thường có nhận đầu vào và có đầu ra:



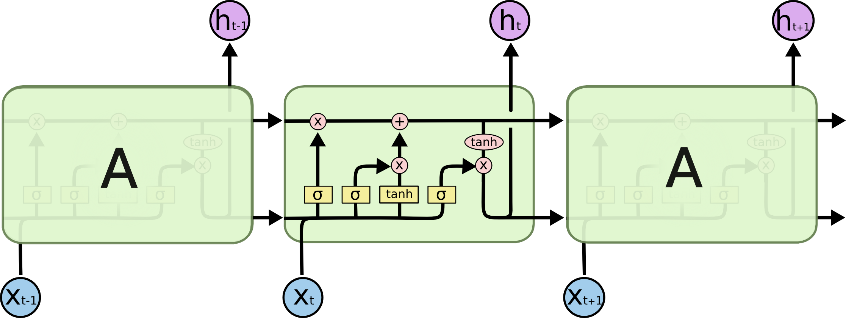
Hình 2.2 Mạng RNN khi được phân giải ra [7]

RNN được gọi là hồi quy vì chúng thực hiện một chuỗi các công việc lặp lại với đầu ra phụ thuộc vào các kết quả trước đó. Trong lĩnh vực sử lý ngôn ngữ tự nhiên mạng RNN đã được áp dụng rất thành công tuy nhiên nếu một văn bản dài và thông tin quan trọng xuất hiện một cách rời rạc làm cho khoảng cách thông tin trở nên xa hơn thì mạng RNN sẽ khó có thể nhớ được nữa được gọi là vấn đề phụ dài hạn.

1. **Mạng Long short-term memory (LSTM)**

Mạng LSTM cũng được xây dựng tương tự như mạng RNN tuy nhiên một đặc điểm của mạng LSTM nổi bật hơn so với mạng RNN truyền thống là việc ghi nhớ được các chi tiết ở xa hơn giúp cho việc dự báo được cải thiện độ chính xác hơn nếu các thông tin quan trọng trong dự báo ở xa hơn thông tin cần dự báo.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn:



Hình 2.1

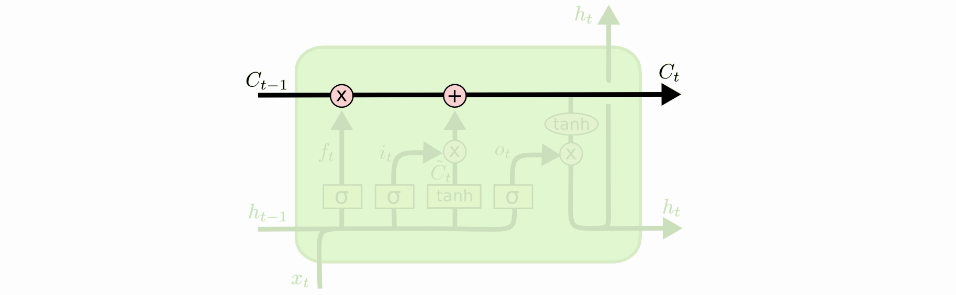
Mạng LSTM (hình 2.1) có kiến trúc dạng chuỗi gồm 4 tầng tương tác với nhau một cách rất đặc biệt



Hình 2.2

Một số kí hiệu cơ bản được sử dụng để mô tả mạng LSTM (hình 2.2) được hiểu như sau: Các ô màu vàng được sử dụng để học trong các tầng mạng nơ-ron, các hình tròn màu hồng biểu diễn toán tử pointwise với các phép toán như phép cộng và phép nhân, mũi tên biểu thị một véc-tơ từ đầu ra của một nút tới đầu vào của một nút khác, các đường hợp nhau kí hiệu việc kết hợp còn các đường rẽ nhánh ám chỉ nội dung của nó được sao chép và chuyển tới các nơi khác nhau.

Cốt lõi của LSTM là là trạng thái nhớ (cell state) đường chạy ngang trong sơ đồ:



Hình 2.3

Nó là một vector chạy xuyên suốt tất cả các mắt xích và chỉ có một vài tương tác tuyến tính rất nhỏ (minor linear interactions). Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi và có thể lọc đi hoặc thêm vào các thông tin cần thiết, được điều chỉnh cẩn thận bởi các nhóm được gọi là “cổng” (gate).

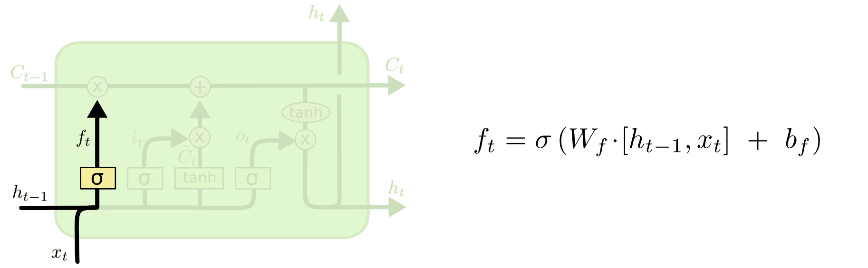
Các cổng là nơi tùy chọn thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một toán tử nhân pointwise.



Hình 2.4 Cổng của mạng LSTM

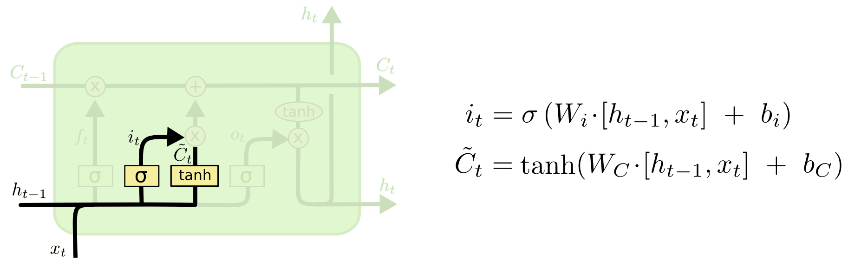
Tầng Sigmoid có giá trị từ 0 đến 1 cho biết mức độ thông tin được truyền qua mạng . Nếu là 0 thì sẽ không cho thông tin nào qua và là 1 thì tất cả thông tin đều được đi qua. Một LSTM gồm 3 cổng như vậy để bảo vệ và điều hành cell state.

Cách hoạt động của mô hình LSTM gồn các bước sau. Đầu tiên là quyết định xem thông tin nào được bỏ vào trạng thái cell nhờ tầng sigmoid - gọi là “forget gate layer” ­- cổng quên. Nó lấy đầu vào là ht-1 và xt đưa ra kết quả 1 là “giữ” 0 là “bỏ” cho mỗi số trong cell state Ct-1.



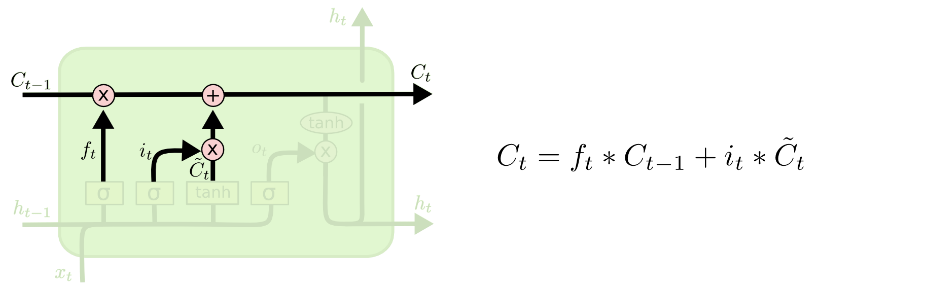
Hình 2.3

Tiếp theo ta cần xác định thông tin mới nào sẽ được lưu vào cell state. Việc này gồm 2 phần. Đầu tiên một tầng sigmoid được gọi là “input gate layer” - cổng vào để quyết định giá trị nào ta sẽ cập nhập. Tiếp theo là một tầng tanh tạo ra một vector cho giá trị mới Ct̃ để thêm vào trạng thái.

****

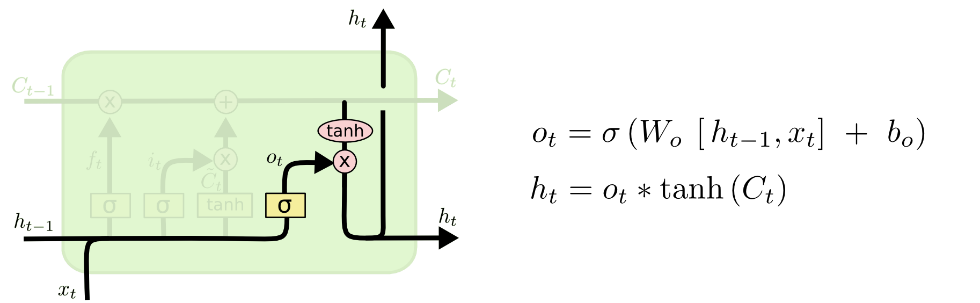
Hình 2.4

Bước tiếp theo ta sẽ cập nhật trạng thái mới từ Ct-1 thành Ct. Ta nhân trạng thái cũ với ft để bỏ đi thông tin muốn quên ở bước đầu. Sau đó cộng thêm it\*Ct̃ để thêm trạng thái mới vào cell.



Hình 2.5

Cuối cùng, ta cần quyết định đầu ra là gì. Giá trị đầu ra sẽ dựa vào trạng thái cell, nhưng sẽ được sàng lọc tiếp tục. Đầu tiên, ta chạy một tầng sigmoid để lọc phần nào của trạng thái tế bào ta muốn xuất ra. Sau đó, trạng thái tế bào đưa qua một hàm tanh để cho giá trị trả về trong khoảng [−1,1], và nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn.



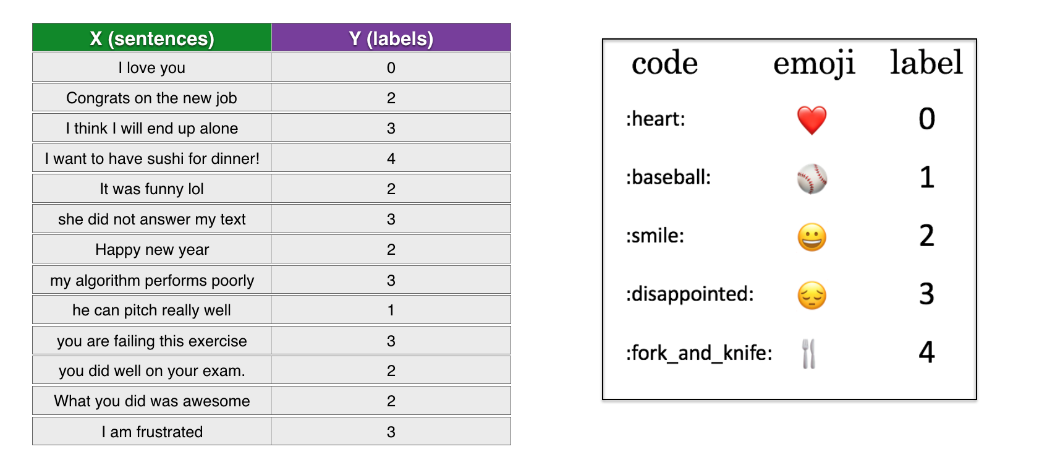
Hình 2.6

**CHƯƠNG 3   
CÀI ĐẶT THỰC NGHIỆM**

1. **Dữ liệu thực nghiệm**

Tập dữ liệu thực nghiệm sẽ bao gồm một tập dữ liệu nhỏ tiếng anh và một tập dữ liệu tổng hợp các bình luận tiếng việt.

Tập dữ liệu tiếng anh được lấy từ một dự án mẫu LSTM phân tích cảm xúc [8] gồm 127 câu và mỗi câu được gán 1 trong 5 nhãn cảm xúc:



Hình

Có tất cả 5 cảm xúc: heart, baseball, smile, disappointed, fork and knife biểu thị bằng 5 nhãn dạng số từ 0 đến 4 được cung cấp dưới dạng file csv khoảng 4,71 KB.

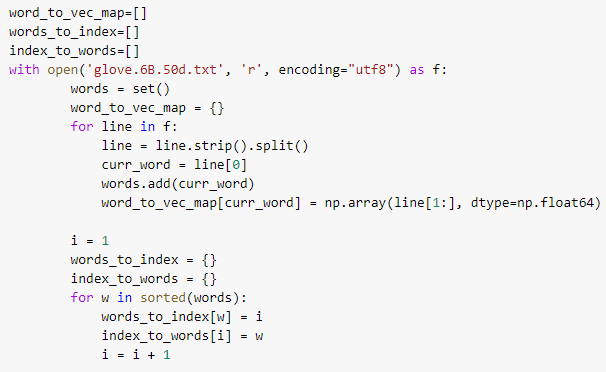
Đối với dữ liệu tiếng việt em sử dụng được lấy từ SCV (streetcodevn) [9] bao gồm bộ dữ liệu tiếng việt tách từ để chuyển đổi từ thành vector và bộ dữ liệu đã tách từ được gán nhãn tích cực và tiêu cực. Cụ thể dữ liệu được cung cấp dưới dạng file text gồm hai thư mục chính là chứa thông tin như sau:

* Data\_train/test: chứa dữ liệu huấn huyện mô hình
* Data\_train/train: dùng để kiểm tra mô hình sau mỗi epoch huấn luyện (validate)
* Data\_test: kiểm tra mô hình sau huấn luyện

Tổng số lượng mẫu của cả hai thư mục là 50.000 mẫu trong đó tập train là 30.000 bình luận, validate là 10.000 bình luận, test là 10.000 bình luận đây là những bình luận từ trang <foody.vn>.

1. **Phân tích cảm xúc với dữ liệu tiếng anh**
2. **Vector hóa văn bản**

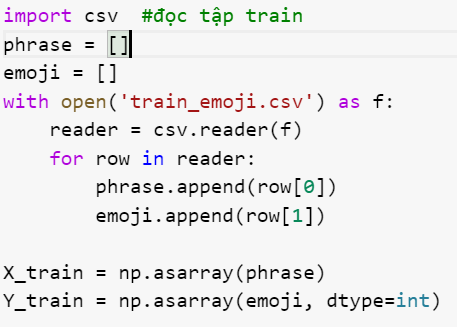
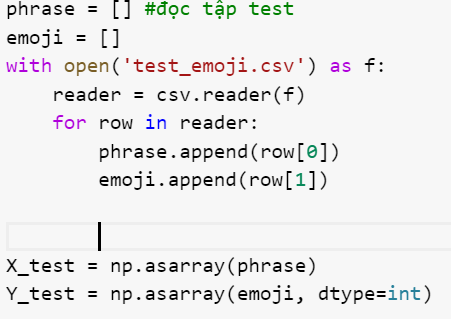
Việc vector hóa văn bản (word embedding) là quá trình tạo các vector cho các từ trong vănn bản để máy tính hiểu được. Có rất nhiều các để vector hóa đã được giớ thiệu ở chương trước một cách đơn giản nhưng hiệu quả mang lại rất tốt là phương pháp GloVe. Trong phương pháp này em sẽ sử dụng các vector đã được huấn luyện và tạo ra sẵn để biêu diễn cho các từ được lấy từ trang Kaggle [10] bao gồm 40.000 từ vựng khác nhau được lưu trong tập text có dung lượng khoảng 168MB và mỗi vector từ sẽ có 50 chiều.



Hình 3.1

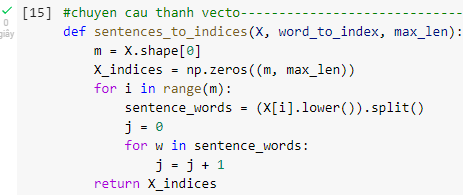
Sau khi đoc file (hình 3.1) dữ liệu ta thu được 3 mảng: chuyển từ thành vectơ (*word\_to\_vec\_map*), chuyển từ thành chỉ số (*words\_to\_index),* chuyển từ chỉ số thành từ (*index\_to\_words*).

Để có được dữ liệu huấn luyện ta sẽ đọc dữ liệu từ hai file train\_emoji.csv (hình 3.2) và test\_emoji.csv (hình 3.3) đây là hai file đã được chia ra từ file chính để làm dữ liệu huấn luyện và kiếm tra. Sau khi hoàn thành ta sẽ có một mảng X\_train,Y\_train, X\_test, Y\_test lần lượt là dữ liệu và nhãn tập train và test.

Hình 3.2 Hình 3.3

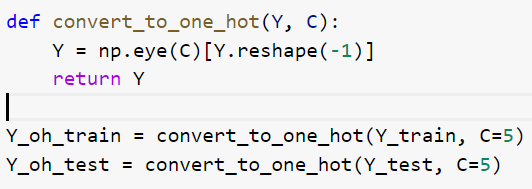
Ta tiếp tục tiến hành chuyển các câu trong tập huấn luyện thành mảng:



Hình 3.2

Hàm *Sentences\_to\_indices()* (hình 3.2) sẽ chuyển câu thành một mảng tương ứng mỗi từ ví dụ: ta có câu “I am a student” thì mỗi từ trong câu tương ứng với một chỉ số trong *words\_to\_index* và câu trên sẽ được chuyển thành một vectơ tương ứng “I am a student” thành mảng [2,4,3,5].

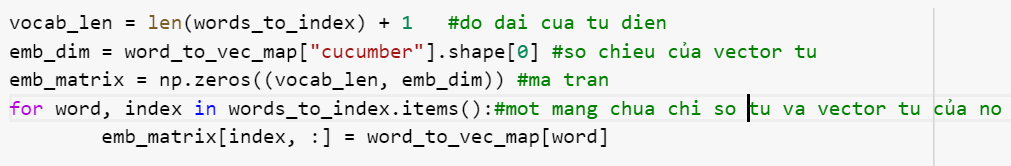
Ta tiến hành chuyển nhãn của từng câu trong tập dữ liệu về vectơ one hot chỉ gồm 0 và 1.



Hình 3.3

Trong bước này mỗi nhãn sẽ chuyển về một mảng một chiều độ dài là 5 chỉ gồm 0 và 1 bằng hàm *convert\_to\_one\_hot*().Ví dụ: [0,1,0,0,0] là biểu diễn của cảm xúc yêu thích.

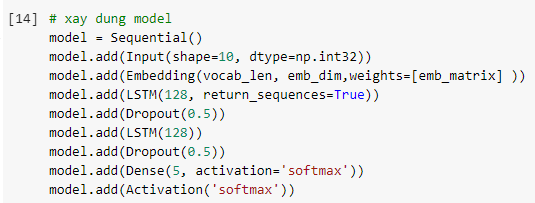
Ta tiếp tục xây dựng một mảng chứa chỉ số của từ đó trong mảng *words\_to\_index* và vector từ tương ứng của từ đó:



Hình 3.4

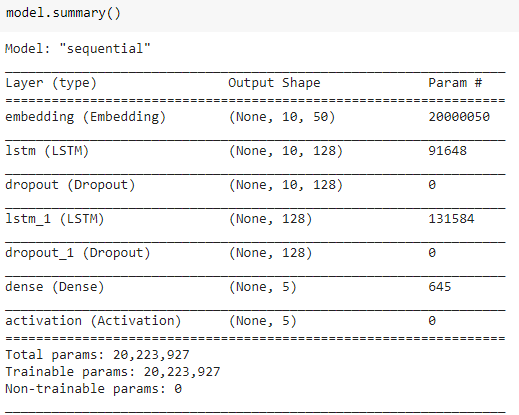
1. **Xây dựng mô hình dự đoán**

Ta tiến hành xây dựng model dự đoán gồm 7 tầng:



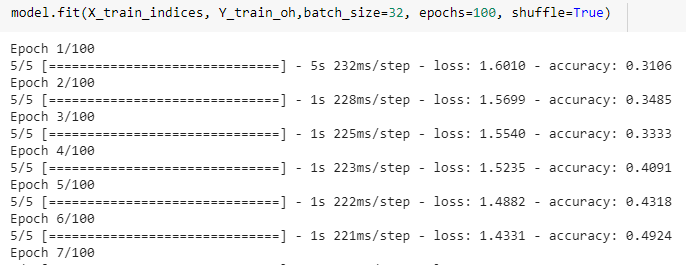
Hình 3.4

Trong model trên (hình 3.4) ta tiến hành thiết lập đầu vào cho mô hình là một mảng bằng 10 phần tử. Sau đó sẽ cho dữ liệu đi qua tầng *Embedding* sử dụng vectơ cho từng từ (*emb\_matrix*) mà ta đã có trước đó đặt vào tầng này. Đầu ra của tầng này sẽ ma trận 10x50 đây chính là vector *Embedding* của mỗi từ sau khi huấn luyện nó giúp các từ nếu có khoảng cách giữa các vectơ là ngắn thì sẽ có đặc điểm tương tự nhau trong câu. Sau khi hoàn thành ta thêm các tầng LSTM và Dropout. Dropout dùng để bỏ qua các đơn vị (units) tức là một nút mạng bất kì trong quá trình huấn luyện để tránh tình trạng học tủ (Over-fitting). Cuối cùng sẽ là tầng Dense một tầng ẩn kết nối các neuron với toàn bộ các node ở phần phía trước nó và nỗi neuron này sẽ có một hàm kích hoạt *softmax* lớp này sẽ thu nhỏ vectơ lại thành một mảng kích thước là 5 tương ứng mỗi phần tử trong mảng là giá trị xác suất của từng nhãn trong kết quả mong đợi.



Hình 3.5

Sau khi xây dựng tổng thể mô hình dự báo (hình 3.5) ta tiến hành cho dữ liệu huấn luyện vào huấn luyện cho mô hình của chúng ta:



Hình 3.6

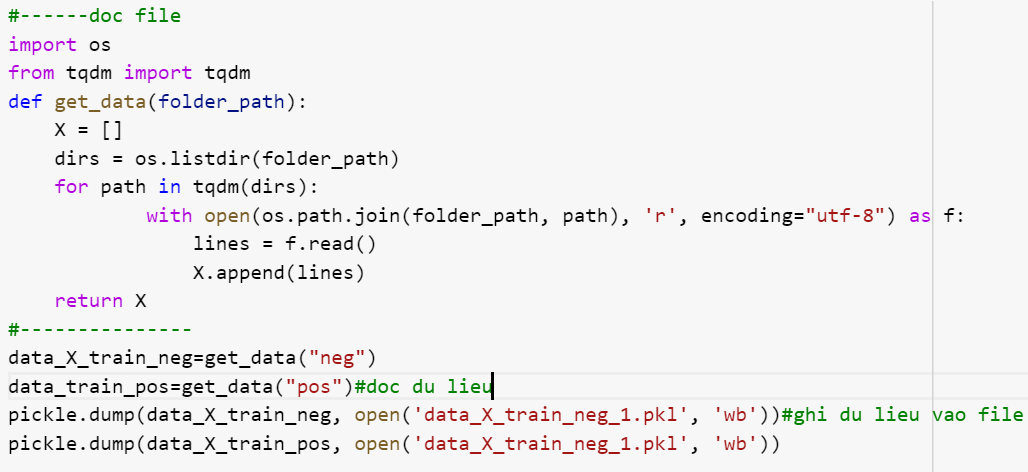
Sau mỗi epoch độ chình xác (accuracy) tăng dần lên cho thấy mô hình đang dự đoán tốt lên nhiều so với trước đó

1. **Phân tích cảm xúc với dữ liệu tiếng Việt**
2. **Tiền xử lý**

Tiền sử lý hay sử lý sơ bộ văn bản là bước đầu tiên trong các mô hình sử lý đầu tiên. Đây là bước rất quan trọng trong xử lý giúp cải thiện kết quả mô hình đáng kể. Một vài tiền sử lý cơ bản được sử dụng thường xuyên là: Chuẩn hóa về chữ thường, tách từ, loại bỏ dấu câu và các ký tự đặc biệt, loại bỏ từ dừng. Tuy nhiên trong bộ dữ liệu huấn luyện này việc tách từ đã được thực hiện sẵn đồng thời việc loại bỏ các kí hiệu đặc biệt cũng được tự động thực hiện trong khi đưa văn bản về dạng vector nên ta có thể bỏ qua bước này hoặc nếu muốn thực hiện việc loại bỏ kí kí hiệu đặc biệt như dấu chấm, dấu phẩy, dấu mở đóng ngoặc,dấu chấm than,... ta có thể dùng thư viện gensim (*gensim.utils.simple\_preprocess*(text)).

1. **Vector hóa văn bản**

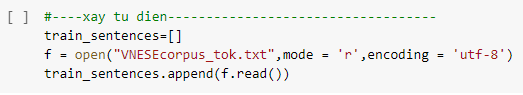
Đầu tiên ta tiến hành đọc file dữ liệu huấn luyện và tập xác minh dữ liệu cho việc huấn luyện mô hình.



Hình 3.7

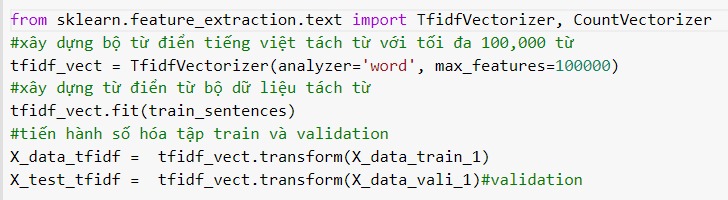
Quá trình đọc và lưu lại thành file.pkl của dữ liệu huấn luyện trong thư mục data\_train\train (Hình 3.7) sẽ giúp cho việc sử dụng lại dữ liệu dễ dàng khi ta phải đọc số lượng file text lên tới hàng ngàn file. Việc đọc và ghi lại file.pkl của tập dữ liệu validate trong thư mục data\_train\test cũng sử dụng tương tự. Sau khi hoàn thành ta sẽ tiếp tục thực hiện tạo một mảng chứa các nhãn tương ứng 1 là tích cực 0 là tiêu cực cho các tập dữ liệu trên để tiến hành thực nghiệm.

Tiến hành xây dựng từ điển bằng cách đọc file bộ dữ liệu tiếng việt tách từ từ trang [https://streetcodevn.com/blog/dataset](https://streetcodevn.com/blog/dataset%20) trong file này là một loạt các câu đã được tách từ sẵn ví dụ: “1 năm bản\_quyền miễn\_phí phần\_mềm bảo\_mật hàng\_đầu” các từ có nghĩ khi kết hợp với nhau được nhóm lại một cụm giúp cho quá trình chuyển từ thành chỉ số được dễ dàng hơn.



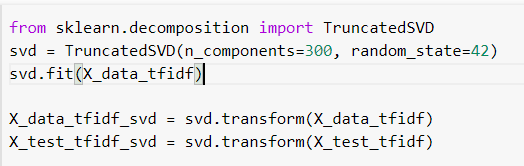
Hình 3.12

Ta tiến hành chuyển đổi dữ liệu có dạng văn bản thành các vector số học. Do ở bộ dữ liệu tiếng Anh ta đã sử dụng phương pháp Neural Embedding (phương pháp GloVe) nên bây giờ ta sẽ dùng một kĩ thuật Word Embedding cổ điển, sử dụng phương pháp tf-idf để số hóa các từ trong văn bản thông qua tần suất suất hiện của chúng.



Hình 3.13

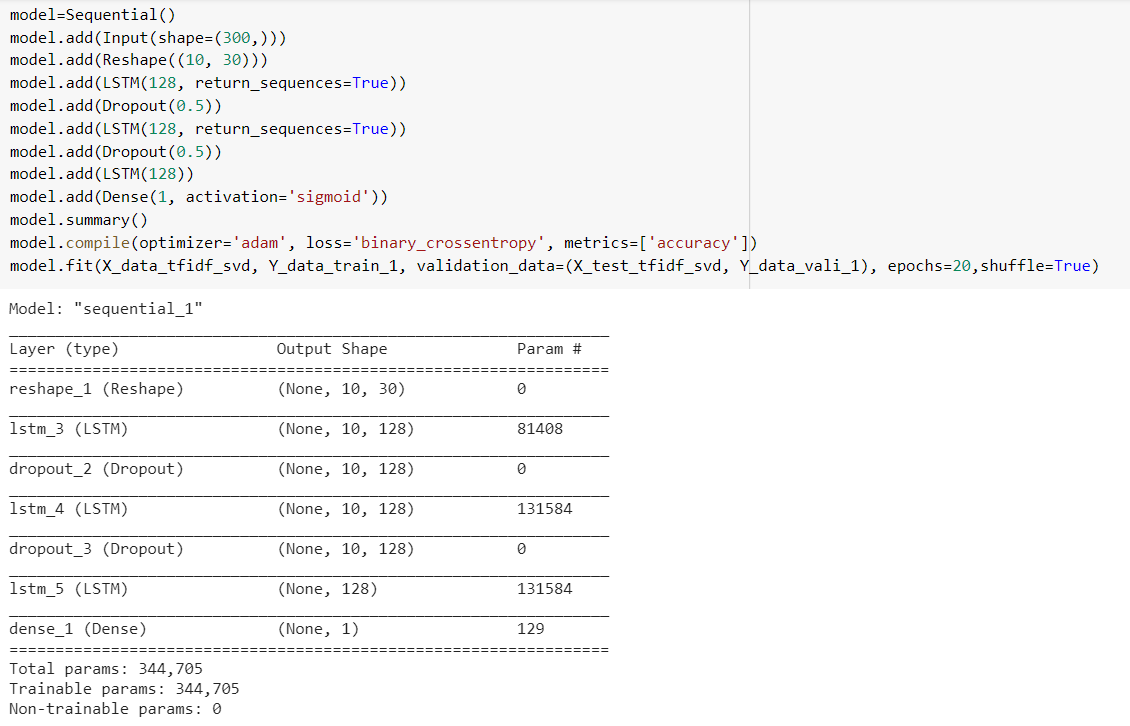
Sau khi thực hiện xong mặc dù ta đã thu được ma trận của các từ nhưng ma trận này lại có kích thước quá lớn việc sử lý tính toán với ma trận này đòi hỏi thời gian và bộ nhớ khá tốn kém để sử lý vấn đề này ta sử dụng thuật toán SVD (singular value decomposition) để giảm chiều của ma trận xuống mà vẫn giữ nguyên được các thuộc tính ban đầu:



Hình 3.14

1. **Xây dựng mô hình dự đoán**

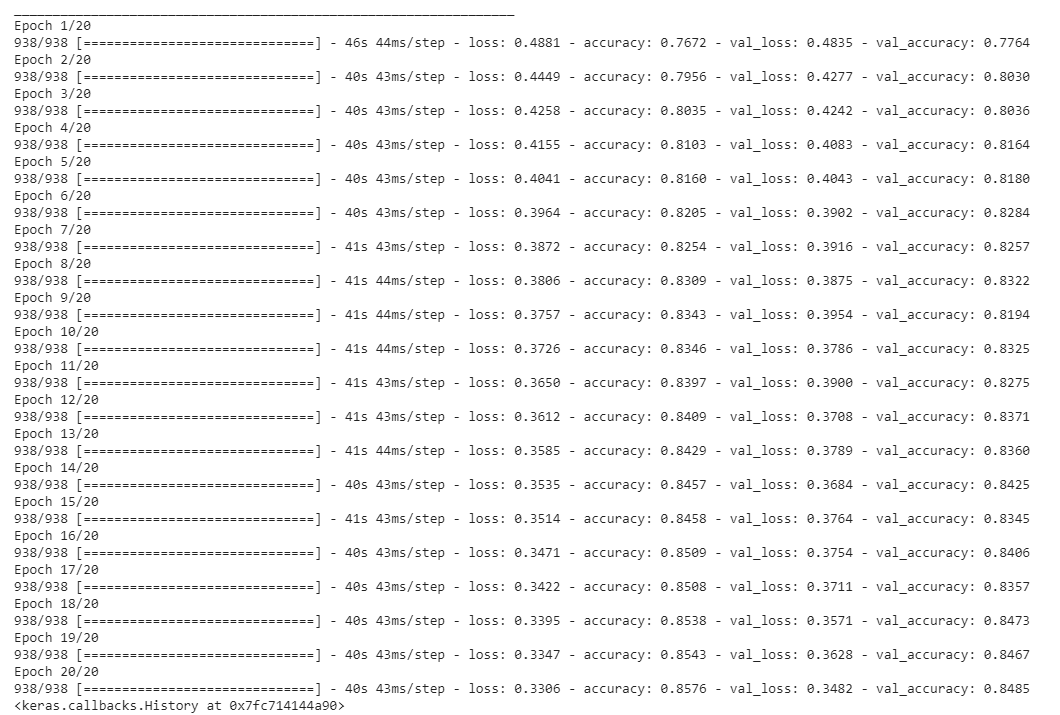
Tiếp theo ta xây dựng model tương tự như phân loại câu tiếng anh tuy nhiên lớp cuối cùng sẽ là lớp *Dense* (hàm kích hoạt *sigmoid* ) đầu ra là 1 thể hiện cho một câu là tích cực nếu đầu ra lớn hơn 0.5 (xác xuất để câu đó là tích cực):



Hình 3.15

Trong mô hình này ta sẽ thay đổi kích thước ma trận đầu vào từ mảng 1 chiều 300 sang mảng mảng 10x30 sau đó tiến hành cho vào mô hình LSTM để huấn luyện.

Sau khi thiết lập xong mô hình ta tiến hành huấn luyện với tập dữ liệu huấn luyện và xác minh mỗi epoch thông quan tập xác minh:



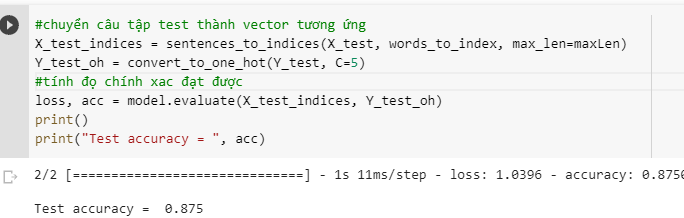
Hình 3.16

Ta có thể thấy sau mỗi epoch độ chính xác sẽ tăng dần từ 77,6% lên 84,9% .

1. **Kết quả dự đoán**

Sau khi huấn luyện song ta tiến hành kiểm tra độ chính xác của mô hình thông quan tập dữ liệu kiểm tra đã chuẩn bị.

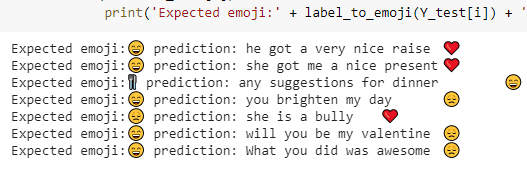
1. **Dữ liệu tiếng Anh**



Hình 3.7

Chuyển các câu trong dữ liệu sang dạng vector, tiến hành dự đoán và so sánh với nhãn thực tế đạt được (Hình 3.7). Sau khi tiến hành dự đoán ta chuyển các kết quả về dạng one-hot vector khi này trong mảng gồm 5 phần tử, phần tử nào lớn nhất tương đương với xác xuất dự đoán là cao nhất sẽ nhật giá trị 1 còn lại các phần tử khác là 0 .Mô hình cho lại độ chình xác rất cao lên đén 87.5% cho tập kiểm tra

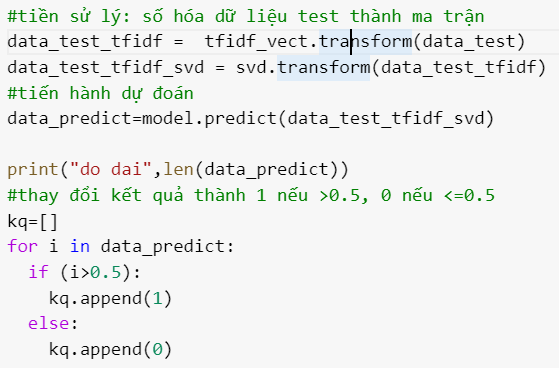
Sau đây là các câu cho kết quả dự đoán sai:



Hình 3.8

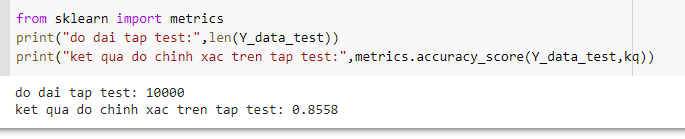
1. **Dữ liệu tiếng Việt**

Sau khi huấn luyện mô hình ta tiến hành số hóa tập dữ liệu kiểm tra gồm 10000 câu bình luận:



Hình 3.17

Chuyển các câu trong dữ liệu sang dạng vector, tiến hành dự đoán và so sánh với nhãn thực tế đạt được (Hình 3.17). Do kết quả của quá trình dự đoán là xác xuất để bình luận là tích cực hay không nên ta sẽ thay đổi kết quả là 1 nếu xác xuất lớn hơn 0.5 là 0 nếu nhỏ hơn. Kiểm tra độ chính xác của mô hình:



Hình 3.18

Độ chính xác của mô hình trên tập test lên đến 85,6% cho thấy mô hình này đã dữ đoán thành công.

1. **r**