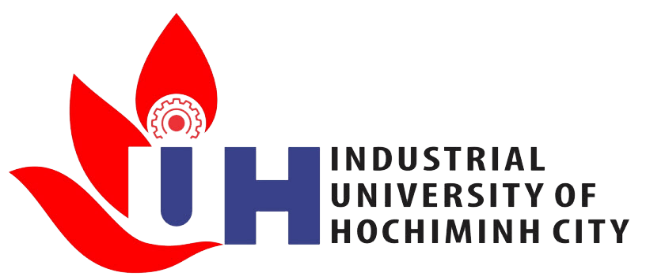
**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH**

**---------------------------**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

**MÔN:** **XỬ LÍ NGÔN NGỮ TỰ NHIÊN**

***Đề tài:***

**PHÂN LOẠI COMMENT**

Lớp học phần: DHKHMT15A

GVHD: Đặng Thị Phúc

|  |  |  |
| --- | --- | --- |
| **STT** | **HỌ VÀ TÊN** | **MSSV** |
| 1 | Trần Đông Hoàng | 18081491 |
|  |  |  |

Thành phố Hồ Chí Minh, tháng 5 năm 2022

Contents

[1. Về bài toán phân loại văn bản: 2](#_Toc103895718)

[2. Quy trình tổng quan hiện thực bài toán phân loại văn bản 2](#_Toc103895719)

[3. Thuật toán sử dụng 3](#_Toc103895720)

[3.1. GRU 3](#_Toc103895721)

[3.2 Bidirectional 6](#_Toc103895722)

[3.3 Mạng CNN 9](#_Toc103895723)

# Về bài toán phân loại văn bản:

Trong Machine Learning và NLP, phân lớp văn bản là một bài toán xử lí văn bản cổ điển, gán các nhãn phân loại lên một văn bản mới dựa trên mức độ tương tự của văn bản đó so với các văn bản đã được gán nhãn trong tập huấn luyện.

Các ứng dụng của phân lớp văn bản thường rất đa dạng như: lọc email spam, phân tích cảm xúc (sentiment analysis), phân loại tin tức, …

# Quy trình tổng quan hiện thực bài toán phân loại văn bản

[](https://github.com/huynhminhtan/thuattoanthongminh/blob/master/assets/computational-biology.png)

Luồng xử lý cơ bản:

**Crawler data** (cào dữ liệu) -> **text normalization** (chuẩn hóa dữ liệu) -> **data preprocessing** (tiền xử lý dữ liệu) -> **features** (trích xuất đặc trưng) -> **learn/train model** (chọn model machine learning và huấn luyện) -> **evaluation/results** (đánh giá kết quả).

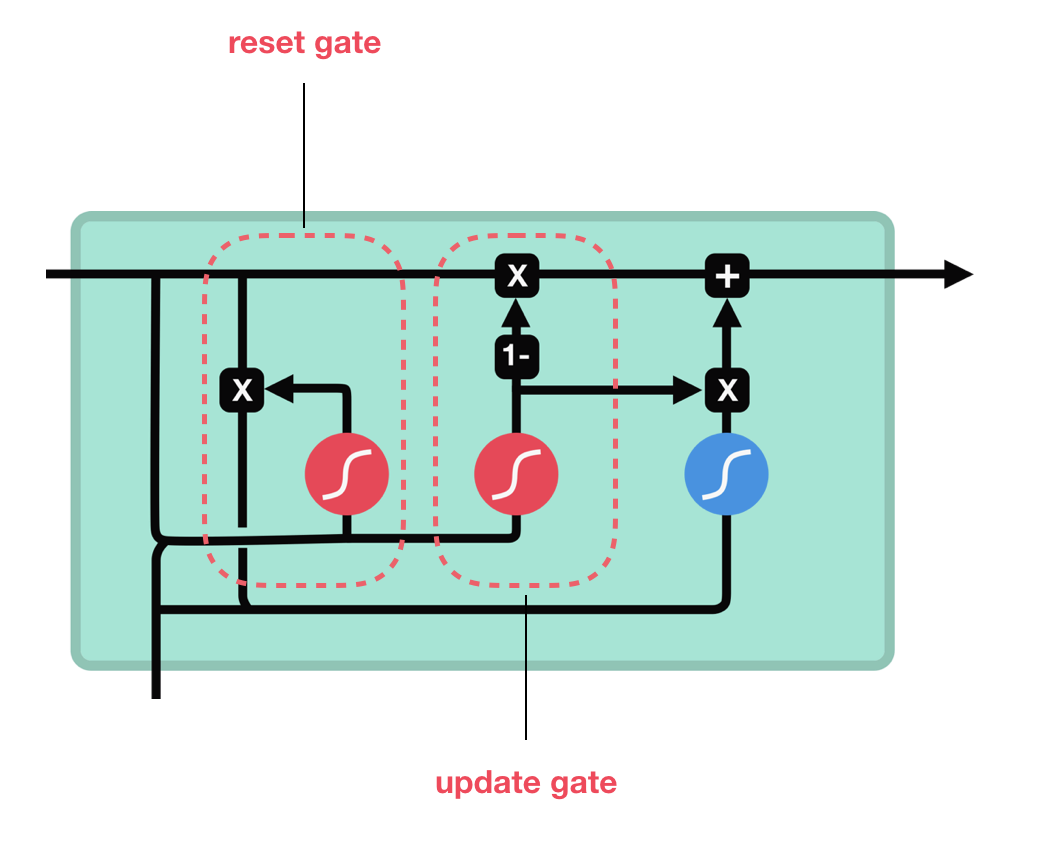
Trong đó:

* **Crawler data** (cào dữ liệu): Là công đoạn chuẩn bị tập dataset (bộ dữ liệu để sử dụng) được lấy trên tiki và lazada gồm có 7000 câu comment tự crawl label thành 12 lớp
* **Text normalization** (chuẩn hóa dữ liệu): Công đoạn loại bỏ các thành phần không cần thiết từ dữ liệu mới crawler được có thể hiểu là làm sạch dữ liệu xóa đi dữ liệu rác cuối cùng nhận được đoạn văn bản chỉ có text. Ví dụ, xóa đi tag HTML, xóa link, xóa ký tự đặc biệt "\n \t &#64",...
* **Data preprocessing** (tiền xử lý dữ liệu): Chuyển dữ liệu/ văn bản nhận được ở giai đoạn trên thành dữ liệu đầu vào (data input) thích hợp cho đúng với mô hình (model machine learning) sử dụng phân loại văn bản. Ví dụ, các công việc cần thực hiện trước khi đưa vào thuật toán phân loại văn bản tiếng Việt như: tách từ, chuẩn hóa từ, loại bỏ stopwords, vertor hóa từ. Đây là công đoạn quan trọng trong bài toán phân loại văn bản.
* **Features** (trích xuất đặc trưng): Với bài toán phân loại trên thực tế, khi muốn phân loại cần phải dựa theo một đặc điểm nào đó như giới tính, hình dạng, kích thước dựa trên sự quan sát hoặc số liệu cụ thể. Trong bài toán phân loại cũng vậy, nhưng nó đòi hỏi việc phải tự động phát hiện ra các đặc điểm của đối tượng rồi mới thực hiện phân loại cho phù hợp. Ví dụ, phân loại hoa Hồng, phải phát hiện ra mỗi hoa đó có đặc điểm như thế nào xét cả về hình dạng, màu, kích thước, giống, mùi hương. Một đối tượng có rất nhiều đặc điểm, vậy dựa trên một hoặc nhiều đặc điểm nào để phân loại? Vì thế công đoạn này sẽ rúc trích hay lựa chọn bộ đặc điểm nào tối ưu nhất, dễ nhận dạng nhất, dễ phát hiện ra đối tượng đó nhất. Cuối cùng công đoạn này sẽ thu được một tập dữ liệu đã được trích xuất sau đó đưa vào thuật toán machine learning phân loại. Có 2 loại feature:
  + Feature Selection (chọn lựa đặc trưng): là chọn ra một tập đặc trưng con từ không gian đặc trưng gốc.
  + Feature Extraction (rút trích đặc trưng): là biến đổi (transform) không gian đặc trưng gốc thành một không gian đặc trưng nhỏ hơn để giảm số chiều đặc trưng. So với phương pháp chọn đặc trưng, rút trích không chỉ giảm số chiều mà còn thành công trong việc giải quyết vấn đề tính nhiều nghĩa (polysemy) và tính đồng nghĩa (synonym) của từ ở mức độ có thể chấp nhận.
* **Learn/train model** (chọn model machine learning và huấn luyện): Lựa chọn một thuật toán tối ưu nhất cho bài toán phân loại văn bản.
* **Evaluation/results** (đánh giá kết quả): Công đoạn cuối, đánh giá kết quả nhận được.

# Thuật toán sử dụng

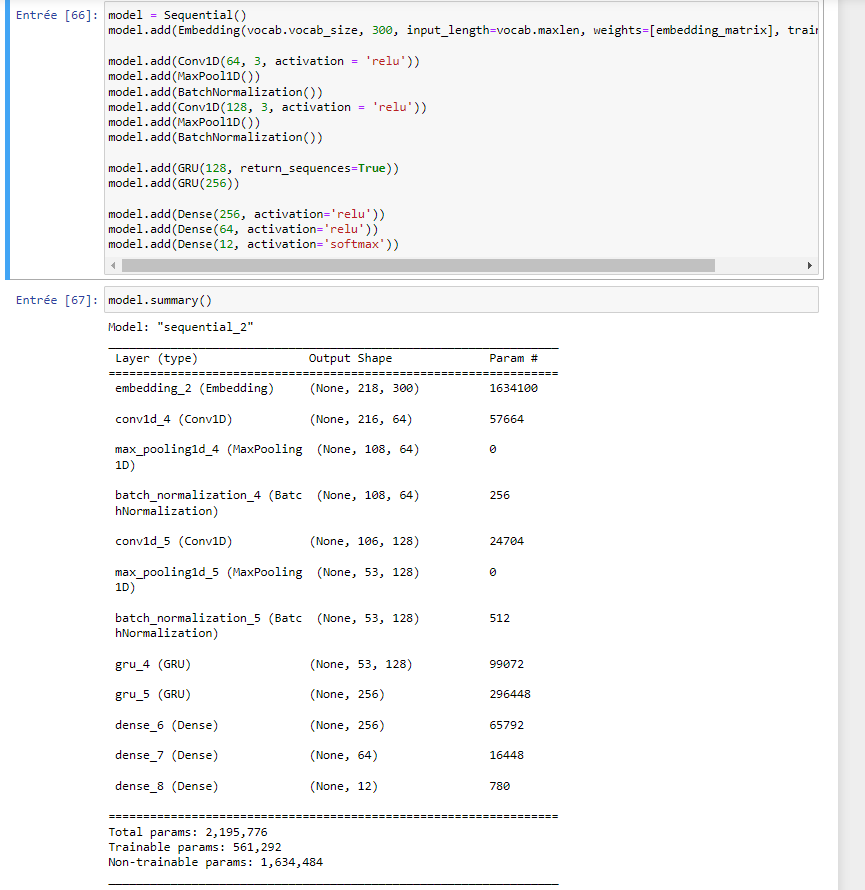
## GRU

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



Do mất đi trạng thái cell, nên tác dụng của 2 cổng trên khá khó để phân biệt rõ ràng, có thể nói là 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.

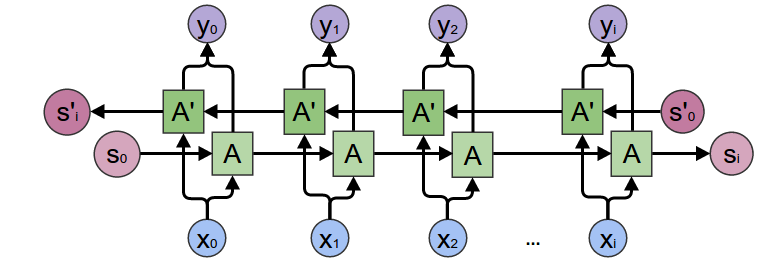
* Model:



* BIểu đồ độ chính xác:
* 

## Bidirectional

Để cho phép truyền dữ liệu đầu vào thẳng (quá khứ) và đảo ngược (tương lai), các RNN hai chiều hoặc BRNN được sử dụng. BRNN là sự kết hợp của hai RNN - một RNN di chuyển về phía trước, bắt đầu từ đầu chuỗi dữ liệu và RNN kia, lùi lại, bắt đầu từ cuối chuỗi dữ liệu. Các khối mạng trong BRNN có thể là RNN, GRU hoặc LSTM đơn giản.



BRNN có thêm một lớp ẩn để phù hợp với quá trình đào tạo ngược. Tại bất kỳ thời điểm nàott, các trạng thái ẩn tiến và lùi được cập nhật như sau:

*At*​(*Forward*)=*ϕ*(*Xt*​∗*WXAforward*​+*At*−1​(*Forward*)∗*WAAforward*​+*bAforward*​)

At(Backward)=ϕ(Xt∗WbackwardXA+At+1(Backward)∗WbackwardAA+bbackwardA)

ở đâuϕϕlà chức năng kích hoạt,WW, ma trận trọng lượng, vàbb, sự thiện vị.

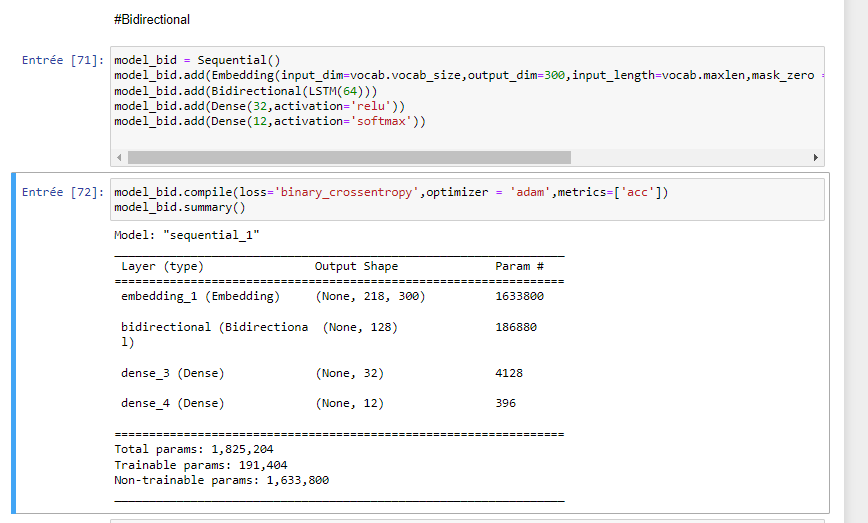
Trạng thái ẩn tại thời điểmttđược đưa ra bởi sự kết hợp At(Forward  
At(Backward). Đầu ra ở bất kỳ trạng thái ẩn nào đã cho là:

*Ot*​=*Ht*​∗*WAY*​+*bY*​  
Việc huấn luyện BRNN tương tự như thuật toán Truyền ngược thời gian (BPTT). BPTT là thuật toán lan truyền ngược được sử dụng trong khi huấn luyện RNN. Một thuật toán BPTT điển hình hoạt động như sau:

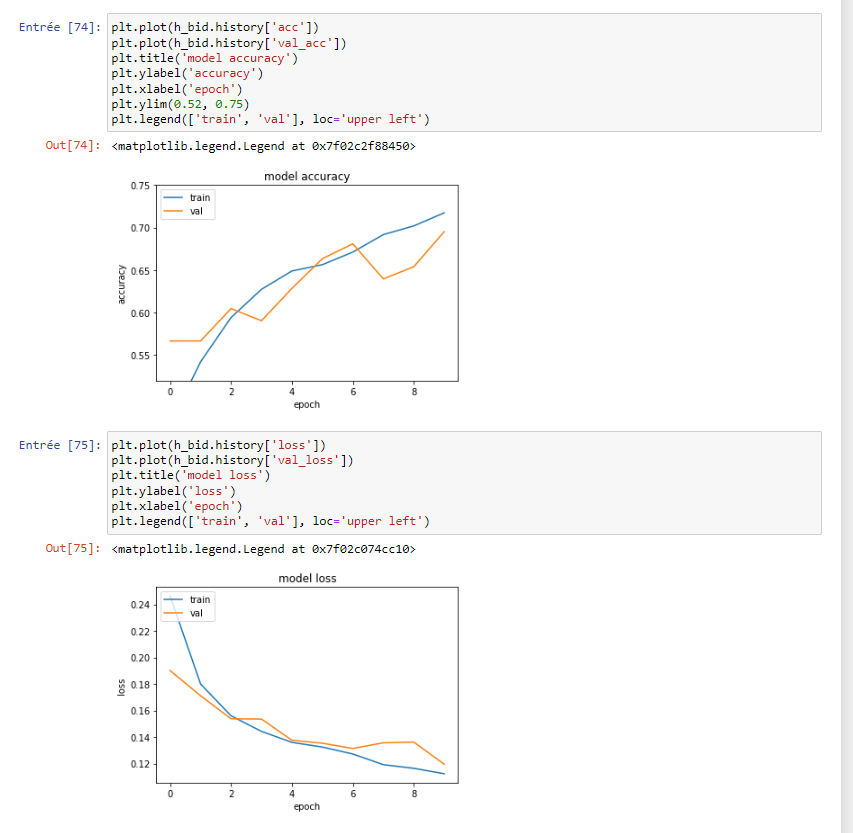
* Hủy đăng ký mạng và tính toán lỗi tại mọi bước.
* Cuộn mạng và cập nhật trọng số.

Tuy nhiên, trong BRNN, vì có quá trình chuyển tiếp và chuyển tiếp xảy ra đồng thời, việc cập nhật trọng số cho hai quá trình có thể xảy ra cùng một thời điểm. Điều này dẫn đến kết quả sai lầm. Do đó, để điều chỉnh các đường chuyền chuyển tiếp và chuyển tiếp lùi một cách riêng biệt, thuật toán sau được sử dụng để đào tạo một BRNN:

* Model:

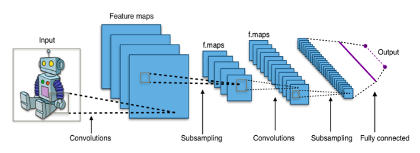


* BIểu đồ độ chính xác:

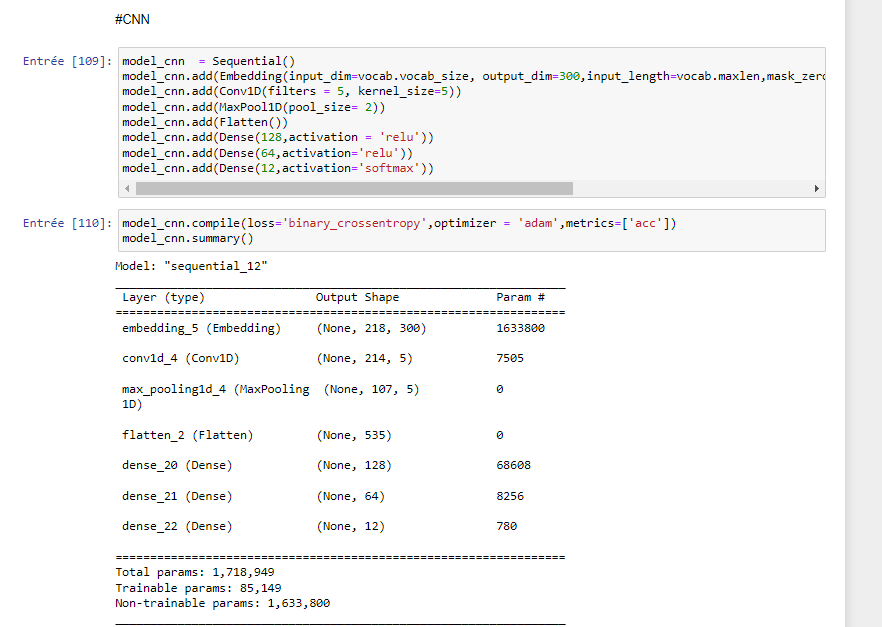


## Mạng CNN

* Mạng CNN là một tập hợp các lớp tích chập chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.
* Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo. Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer). Còn trong mô hình CNNs thì ngược lại. Các lớp liên kết được với nhau thông qua cơ chế tích chập. Lớp tiếp theo là kết quả phép tính tích chập từ lớp trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó. Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số lớp khác như pooling/subsampling lớp dùng để chắt lọc lại các thông tin hữu ích hơn. Trong quá trình huấn luyện mạng (traning) CNN tự động học các giá trị qua các lớp 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. Lớp cuối cùng được dùng để phân lớp ảnh.



* Model:



* BIểu đồ độ chính xác:

