**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**



BÁO CÁO THUỘC HỌC PHẦN: TRÍ TUỆ NHÂN TẠO

**Đề tài: NGHIÊN CỨU VỀ MẠNG NEURAL NHÂN TẠO VÀ ỨNG DỤNG TRONG CHATBOT**

|  |  |
| --- | --- |
| GVHD: | TS. Vũ Thanh Hiền |
| Lớp: | 19DTHD2 |
| Thành viên: | Huỳnh Nhựt Quang |
|  | Lê Thị Thanh Hoa |
|  | Trần Tấn Phát |
|  | Lê Hoàn Phúc |

Hồ Chí Minh, Năm 2021

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tuần** | **Người thực hiện** | **Nội dung công việc** | **Kết quả đạt được** | **Đánh giá** |
| 1 | Huỳnh Nhựt Quang  Lê Hoàn Phúc  Lê Thị Thanh Hoa | Đề xuất đề tài và lên ý tưởng, tổng hợp các công việc, lập kế hoạch chi tiết | Lập được kế hoạch thực hiện đề tài,  Tìm tài liệu tham khảo | 20% |
| 2 | Lê Thị Thanh Hoa  Trần Tấn Phát | Phân công công việc cụ thể cho từng thành viên | Phân chia công việc cho từng thành viên thực hiện | 20% |
| 3 | Lê Hoàn Phúc  Huỳnh Nhựt Quang  Lê Thị Thanh Hoa  Trần Tấn Phát | Tìm hiểu về một số mẫu chatbot thông dụng | Cơ bản nắm được những chức năng và giao diện của một chương trình chuẩn, tham khảo những chường trình trên thị trường | 20% |
| 4 | Trần Tấn Phát  Lê Thị Thanh Hoa  Lê Hoàn Phúc | Soạn báo cáo  (word, Powerpoint) | Tổng hợp nội dung , hoàn thành báo cáo môn học | 20% |
| 5 | Huỳnh Nhựt Quang  Lê Thị Thanh Hoa  Trần Tấn Phát  Lê Hoàn Phúc | Thực hiện nghiên cứu về mạng neuron nhân tạo và ứng dụng trong chat bot | Nắm được kiến thức về mạng neuron nhân tạo, ứng dụng và thực hiện vào việc tạo Chat Bot bằng neuron nhân tạo | 20% |

**BẢNG PHÂN CÔNG CÔNG VIỆC**

**MỤC LỤC**

[Chương 1. Mạng Neural nhân tạo 1](#_Toc86590381)

[1.1 Kiến trúc mạng Neural nhân tạo 1](#_Toc86590383)

[Hình 1.1.1: Nơ ron ở người 1](#_Toc86590384)

[Hình 1.1.2: Kiến trúc mạng nơ ron nhân tạo 2](#_Toc86590385)

[Hình 1.1.3: Quá trình xử lý thông tin của một mạng nơ-ron nhân tạo 3](#_Toc86590386)

[Hình 1.1.4: **Summation Function** (Hàm tổng) 3](#_Toc86590387)

[Hình 1.1.5: Sigmoid hoặc Tanh. 4](#_Toc86590388)

[Hình 1.1.6: Lan truyền thẳng (forward propagation). 5](#_Toc86590389)

[Hình 1.1.7: crossentropy loss (còn gọi là negative log likelihood) 5](#_Toc86590390)

[1.2 Mạng nơ ron hồi quy RNN 5](#_Toc86590391)

[Hình 1.2.1: Mạng RNN 6](#_Toc86590392)

[1.3 Mạng Long Short Term Memory (LSTM) 6](#_Toc86590393)

[Hình 1.3.1: Các mô-đun lặp của mạng LSTM chứa bốn layer. 7](#_Toc86590394)

[Chương 2. Ứng dụng trong Chatbot 8](#_Toc86590396)

[2.1 Những ứng dụng của mạng nơ ron trong chatbot 8](#_Toc86590397)

[2.2 Mạng nơ ron ứng dụng trong Chatbot 8](#_Toc86590398)

[Hình 2.2.1: Sơ đồ cấu trúc hệ thống. 8](#_Toc86590399)

[Hình 2.2.2:Chương trình code. 10](#_Toc86590400)

[2.3 Demo sản phẩm 11](#_Toc86590401)

[Hình 2.3.1:Giao diện chính của chương trình 11](#_Toc86590402)

[Kết luận 12](#_Toc86590403)

[Tài liệu tham khảo 12](#_Toc86590404)

**MỤC LỤC HÌNH ẢNH**

Hình 1.1.1: Nơ ron ở người 1

Hình 1.1.2: Kiến trúc mạng nơ ron nhân tạo 2

Hình 1.1.3: Quá trình xử lý thông tin của một mạng nơ-ron nhân tạo 3

Hình 1.1.4: **Summation Function** (Hàm tổng) 3

Hình 1.1.5: Sigmoid hoặc Tanh. 4

Hình 1.1.6: Lan truyền thẳng (forward propagation). 5

Hình 1.1.7: crossentropy loss (còn gọi là negative log likelihood) 5

Hình 1.2.1: Mạng RNN 6

Hình 1.3.1: Các mô-đun lặp của mạng LSTM chứa bốn layer. 7

Hình 2.2.1: Sơ đồ cấu trúc hệ thống. 8

Hình 2.2.2:Chương trình code. 10

Hình 2.3.1:Giao diện chính của chương trình 11

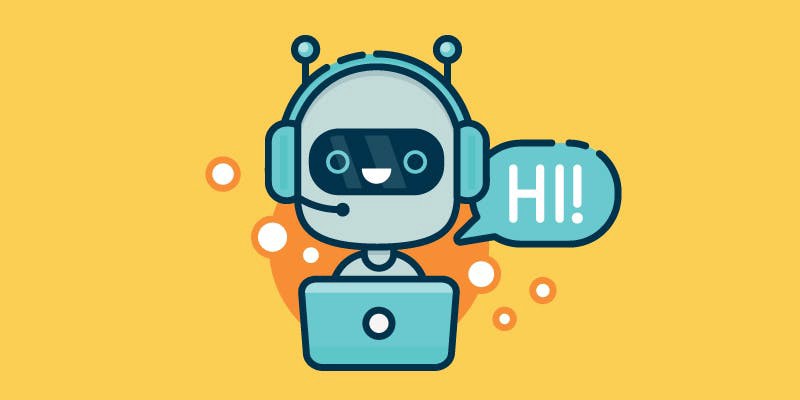
**DANH MỤC CỤM TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Cụm từ** | **Viết tắt** |
| 1 | Artificial Intelligence | AI |
| 2 | Artificial Neural Network | ANN |
| 3 | Long short-term memory | LSTM |
| 4 | Machine Learning | ML |
| 5 | Recurrent Neural Network | RNN |

**MỞ ĐẦU**

*Hiện nay, các ứng dụng trò chuyện trực tuyến được mọi người sử dụng đang bắt đầu trở thành một phương tiện ưa thích để giao tiếp với các doanh nghiệp và giải quyết thắc mắc của khách hàng. Ứng dụng nhắn tin nhanh đã trở thành điểm đến hàng đầu cho mọi thương hiệu nhằm tiếp cận người tiêu dùng. Chính vì vậy, nhu cầu cấp thiết là cần một hệ thống điều khiển thông minh, tự động để mang lại hiệu quả cao hơn và Chatbot là một sự lựa chọn hoàn hảo.*

Một trong số các thành phần cơ bản và quan trọng nhất để tạo ra một chatbot là Artificial **neural networks** (ANN). Chúng dựa trên mô hình hoạt động của các tế bào thần kinh và khớp thần kinh trong não của con người để mô phỏng và tạo ra một bộ não nhân tạo. Bộ não nhân tạo này có vai trò trong việc xử lý ngôn ngữ, xác định ý định người dùng và đưa ra phương án xử lý có độ chính xác cao.

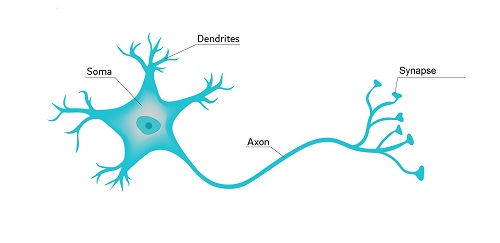


Chương 1. Mạng Neural nhân tạo



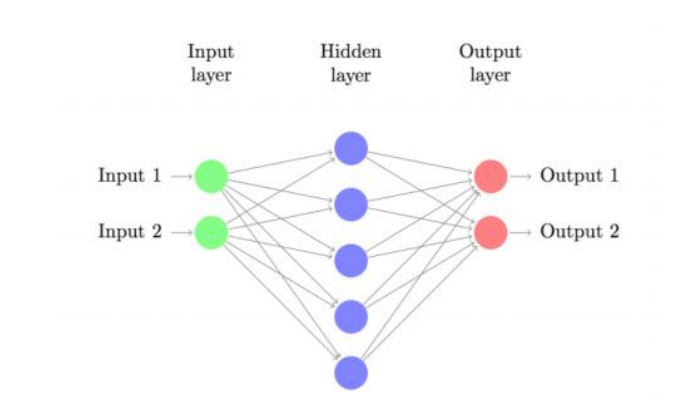
Kiến trúc mạng Neural nhân tạo

Mạng nơ ron nhân tạo (**Artificial Neural Network – ANN**) là một mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh của sinh vật, bao gồm số lượng lớn các Nơ-ron được gắn kết để xử lý thông tin. ANN hoạt động giống như bộ não của con người, được học bởi kinh nghiệm (thông qua việc huấn luyện), có khả năng lưu giữ các tri thức và sử dụng các tri thức đó trong việc dự đoán các dữ liệu chưa biết (unseen data).



Hình 1.1.1: Nơ ron ở người

Một mạng nơ-ron là một nhóm các nút nối với nhau, mô phỏng mạng nơ-ron thần kinh của não người. Mạng nơ ron nhân tạo được thể hiện thông qua ba thành phần cơ bản: mô hình của nơ ron, cấu trúc và sự liên kết giữa các nơ ron. Trong nhiều trường hợp, mạng nơ ron nhân tạo là một hệ thống thích ứng, tự thay đổi cấu trúc của mình dựa trên các thông tin bên ngoài hay bên trong chạy qua mạng trong quá trình học.



Hình 1.1.2: Kiến trúc mạng nơ ron nhân tạo

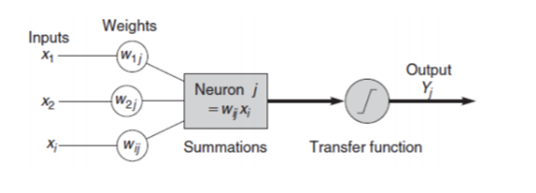
Kiến trúc chung của một ANN gồm 3 thành phần đó là Input Layer, Hidden Layer và Output Layer

Trong đó, lớp ẩn (Hidden Layer) gồm các nơ-ron, nhận dữ liệu input từ các Nơron ở lớp (Layer) trước đó và chuyển đổi các input này cho các lớp xử lý tiếp theo. Trong một mạng ANN có thể có nhiều Hidden Layer.

Lợi thế lớn nhất của các mạng ANN là khả năng được sử dụng như một cơ chế xấp xỉ hàm tùy ý mà “học” được từ các dữ liệu quan sát. Tuy nhiên, sử dụng chúng không đơn giản như vậy, một số các đặc tính và kinh nghiệm khi thiết kế một mạng nơ-ron ANN.

Phương pháp này là tính toán tỷ lệ chính xác dữ liệu đầu ra (output) từ dữ liệu đầu vào (input) bằng cách tính toán các trọng số cho mỗi kết nối (connection) từ các lần lặp lại trong khi “huấn luyện” dữ liệu cho Chatbot. Mỗi bước “huấn luyện” dữ liệu cho Chatbot sẽ sửa đổi các trọng số dẫn đến dữ liệu output được xuất ra với độ chính xác cao.

Thuật toán học: Có hai vấn đề cần học đối với mỗi mạng ANN, đó là học tham số của mô hình (parameter learning) và học cấu trúc (structure learning). Học tham số là thay đổi trọng số của các liên kết giữa các nơ-ron trong một mạng, còn học cấu trúc là việc điều chỉnh cấu trúc mạng bằng việc thay đổi số lớp ẩn, số nơ-ron mỗi lớp và cách liên kết giữa chúng. Hai vấn đề này có thể được thực hiện đồng thời hoặc tách biệt. Nếu các mô hình, hàm chi phí và thuật toán học được lựa chọn một cách thích hợp, thì mạng ANN sẽ cho kết quả có thể vô cùng mạnh mẽ và hiệu quả.



Hình 1.1.3: Quá trình xử lý thông tin của một mạng nơ-ron nhân tạo

**Inputs:** Mỗi Input tương ứng với 1 đặc trưng của dữ liệu. Ví dụ như trong ứng dụng của ngân hàng xem xét có chấp nhận cho khách hàng vay tiền hay không thì mỗi input là một thuộc tính của khách hàng như thu nhập, nghề nghiệp, tuổi, số con,…

**Output:** Kết quả của một ANN là một giải pháp cho một vấn đề.

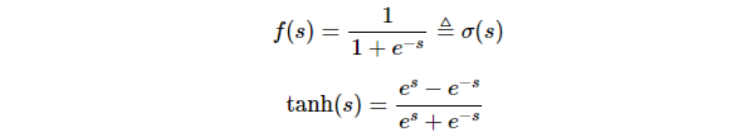
**Connection Weights** (Trọng số liên kết) **:** Đây là thành phần rất quan trọng của một ANN, nó thể hiện mức độ quan trọng, độ mạnh của dữ liệu đầu vào đối với quá trình xử lý thông tin chuyển đổi dữ liệu từ Layer này sang layer khác. Quá trình học của ANN thực ra là quá trình điều chỉnh các trọng số Weight của các dữ liệu đầu vào để có được kết quả mong muốn.

**Summation Function** (Hàm tổng)**:** Tính tổng trọng số của tất cả các input được đưa vào mỗi Nơ-ron. Hàm tổng của một Nơ-ron đối với n input được tính theo công thức sau:

Hình 1.1.4: **Summation Function** (Hàm tổng)

**Transfer Function** (Hàm chuyển đổi): Hàm tổng của một nơ-ron cho biết khả năng kích hoạt của nơ-ron đó còn gọi là kích hoạt bên trong. Các nơ-ron này có thể sinh ra một output hoặc không trong mạng ANN, nói cách khác rằng có thể output của 1 Nơ-ron có thể được chuyển đến layer tiếp trong mạng Nơ-ron theo hoặc không. Mối quan hệ giữa hàm tổng và kết quả output được thể hiện bằng hàm chuyển đổi.

Việc lựa chọn hàm chuyển đổi có tác động lớn đến kết quả đầu ra của mạng ANN. Hàm chuyển đổi phi tuyến được sử dụng phổ biến trong mạng ANN là sigmoid hoặc tanh.

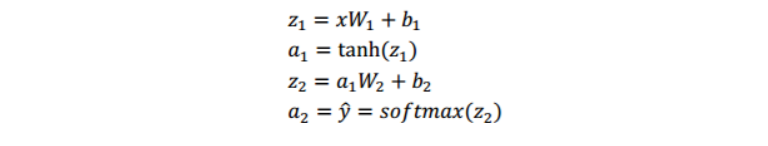


Hình 1.1.5: Sigmoid hoặc Tanh.

Trong đó, hàm tanh là phiên bản thay đổi tỉ lệ của sigmoid , tức là kho tảng giá trị đầu ra của hàm chuyển đổi thuộc khoảng [-1, 1] thay vì [0,1] của Sigmoid nên chúng còn gọi là hàm chuẩn hóa (Normalized Function).

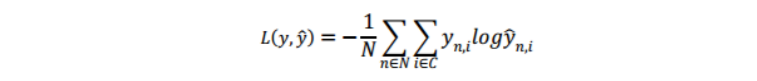
Kết quả xử lý tại các nơ-ron (Output) đôi khi rất lớn, vì vậy hàm chuyển đổi được sử dụng để xử lý output này trước khi chuyển đến layer tiếp theo. Đôi khi thay vì sử dụng Transfer Function người ta sử dụng giá trị ngưỡng (Threshold value) để kiểm soát các output của các neuron tại một layer nào đó trước khi chuyển các output này đến các Layer tiếp theo. Nếu output của một neuron nào đó nhỏ hơn Threshold thì nó sẻ không được chuyển đến Layer tiếp theo.

Mạng nơ-ron của chúng ta dự đoán dựa trên lan truyền thẳng (forward propagation) là các phép nhân ma trận cùng với activation function để thu được kết quả đầu ra. Nếu input x là vector 2 chiều thì ta có thể tính kết quả dự đoán 𝑦 bằng công thức :



Hình 1.1.6: Lan truyền thẳng (forward propagation).

Trong đó, 𝑧𝑖 là input của layer thứ 𝑖, 𝑎𝑖 là output của layer thứ 𝑖 sau khi áp dụng activation function. 𝑊1, 𝑏1, 𝑊2, 𝑏2 là các tham số (parameters) cần tìm của mô hình mạng nơ-ron. Huấn luyện để tìm các tham số cho mô hình tương đương với việc tìm các tham số 𝑊1, 𝑏1, 𝑊2, 𝑏2 sao cho hàm lỗi của mô hình đạt được là thấp nhất. Ta gọi hàm lỗi của mô hình là loss function. Đối với softmax function, ta dùng crossentropy loss (còn gọi là negative log likelihood). Nếu ta có N ví dụ dữ liệu huấn luyện, và C nhóm phân lớp, khi đó hàm lỗi giữa giá trị dự đoán 𝑦 và ŷ được tính:

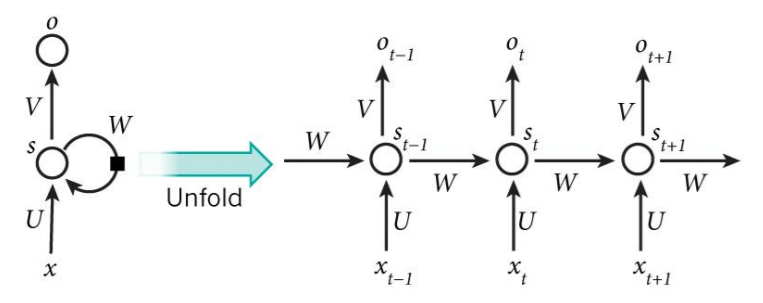


Hình 1.1.7: crossentropy loss (còn gọi là negative log likelihood)

Ý nghĩa công thức trên nghĩa là: lấy tổng trên toàn bộ tập huấn luyện và cộng dồn vào hàm loss nếu kết quả phân lớp sai. Độ dị biệt giữa hai giá trị 𝑦 và ŷ càng lớn thì độ lỗi càng cao. Mục tiêu của chúng ta là tối thiểu hóa hàm lỗi này.

Mạng nơ ron hồi quy RNN

Ý tưởng của RNN đó là thiết kế một Neural Network sao cho có khả năng xử lý được thông tin dạng chuỗi (sequential information), ví dụ một câu là một chuỗi gồm nhiều từ. Trong đó, kết quả đầu ra tại thời điểm hiện tại phụ thuộc vào kết quả tính toán của các thành phần ở những thời điểm trước đó. Nói cách khác, RNN là một mô hình có trí nhớ (memory), có khả năng nhớ được thông tin đã tính toán trước đó.



Hình 1.2.1: Mạng RNN

Mô hình trên mô tả phép triển khai nội dung của một RNN. Triển khai ở đây có thể hiểu đơn giản là ta vẽ ra một mạng nơ-ron chuỗi tuần tự, ví dụ đơn giản trong mạng RNN: X = “How are you?” x1 = How x2 = are x3 = you. Công việc tính toán các state sẽ được thực hiện như sau: S1 = ƒ( S0 , x1 ) S¬2 = ƒ( S1 , x2 ) =ƒ (ƒ ( S0 , x1 ) , x2 ) S3 = ƒ( S2 , x3 ) =ƒ( ƒ( ƒ( S0 , x1 ) , x2 ) , x3 ) Có thể thấy S3 đã mang được thông tin của x1, x2, x3 theo, dễ thấy S3 có thể đại diện cho toàn bộ thông tin của câu How are you.

xt : là đầu vào tại bước t. Ví dụ, x1 là một vec-tơ one-hot tương ứng với từ thứ 2 của câu.

st : là trạng thái ẩn tại t. Nó chính là bộ nhớ của mạng. st được tính toán dựa trên cả các trạng thái ẩn phía trước và đầu vào tại bước đó:st = f(Uxt+Wst−1). Hàm f thường là một hàm phi tuyến tính như tang hyperbolic (tanh) hay ReLu. Để làm phép toán cho phần tử ẩn đầu tiên ta cần khởi tạo thêm s−1, thường giá trị khởi tạo được gắn bằng 0.

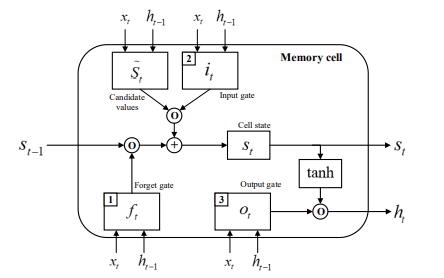
ot : là đầu ra tại bước t. Ví dụ, ta muốn dự đoán từ tiếp theo có thể xuất hiện trong câu thì ot chính là một vectơ xác xuất các từ trong danh sách từ vựng của ta: ot = softmax(Vst) Trong vài năm qua, các nhà nghiên cứu đã phát triển nhiều loại mạng RNNs ngày càng tinh vi để giải quyết các mặt hạn chế của RNN.

Mạng Long Short Term Memory (LSTM)

Long Short Term Memory network (LSTM) là trường hợp đặc biệt của RNN, có khả năng học long-term dependencies. **LSTM** được thiết kế nhằm loại bỏ vấn đề phụ thuộc quá dài.

Cơ chế hoạt động của LSTM là chỉ ghi nhớ những thông tin liên quan, quan trọng cho việc dự đoán, còn các thông tin khác sẽ được bỏ đi.

**LSTM** cũng có cấu trúc mắt xích tương tự, nhưng các module lặp có cấu trúc khác hẳn. Thay vì chỉ có một layer neural network, thì LSTM có tới bốn layer, tương tác với nhau theo một cấu trúc cụ thể.



Hình 1.3.1: Các mô-đun lặp của mạng LSTM chứa bốn layer.

Ý tưởng của LSTM là bổ sung thêm trạng thái bên trong tế bào (cell internal state) 𝑠𝑡 và ba cổng sàng lọc thông tin đầu vào và đầu ra cho tế bào bao gồm cổng quên 𝑓 , cổng đầu vào 𝑖𝑡 và cổng đầu ra 𝑜𝑡 . Tại mỗi bước thời gian t, các cổng lần lượt nhận giá trị đầu vào 𝑥𝑡 (đại diện cho một phần tử trong chuỗi đầu vào) và giá trị ℎ𝑡−1 có được từ đầu ra của các ô nhớ từ bước thời gian trước đó t – 1. Các cổng đều có chức năng sàng lọc thông tin với mỗi mục đích khác nhau. LTSM gồm ba cổng sàng lọc thông tin đầu vào và đầu ra cho tế bào bao gồm cổng quên , cổng đầu vào và cổng đầu ra, các cổng được định nghĩa như sau:

Cổng quên: Có chức năng loại bỏ những thông tin không cần thiết nhận được khỏi trạng thái tế bào bên trong.

Cổng đầu vào: Giúp sàng lọc những thông tin cần thiết để được thêm vào trạng thái tế bào bên trong.

Cổng đầu ra: Có chức năng xác định những thông tin nào từ các trạng thái tế bào bên trong được sử dụng như đầu ra.



Chương 2. Ứng dụng trong Chatbot

Những ứng dụng của mạng nơ ron trong chatbot

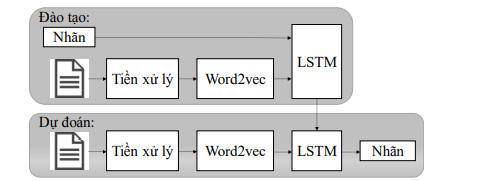
Muốn xây dựng chatbot, Thì hiện nay ta có hai phương pháp phổ biến là phương pháp pipeline và phương pháp end-to-end.

Trong phương pháp pipeline, toàn bộ hệ thống chatbot được chia thành nhiều công đoạn, và thành phần khác nhau, chẳng hạn như đọc hiểu ngôn ngữ tự nhiên (Natural Language Understanding – NLU), quản lý hội thoại (Dialog Management) bao gồm theo dõi trạng thái hội thoại (Dialog State Tracking – DST) học các nguyên lý hội thoại (Policy Learning); sinh ngôn ngữ tự nhiên (Natural Language Generation – NLG). Ngược lại với các phương pháp tiếp cận theo pipeline, các phương pháp end-to-end xây dựng một hệ thống hội thoại bằng cách sử dụng một mô hình duy nhất, trong đó ngữ cảnh ngôn ngữ tự nhiên được lấy làm đầu vào và phản hồi ngôn ngữ tự nhiên được tạo ra làm đầu ra.

Rõ ràng, các hệ thống pipeline với cấu trúc mô-đun dễ hiểu và ổn định hơn, do đó được ứng dụng trong thương mại hóa nhiều hơn. Tuy nhiên, với việc yêu cầu ít chú thích hơn, các hệ thống end-to-end có thể trở thành một giải pháp thay thế đầy hứa hẹn trong tương lai.

Mạng nơ ron ứng dụng trong Chatbot

Nhóm em sử dụng mô hình LSTM vào việc phân loại văn bản cho chatbot, Việc phân loại văn bản là công việc cơ bản và cũng có vai trò quan trọng trong việc nghiên cứu, khai thác lượng dữ liệu lớn hiện nay. Trong bài viết này, chúng tôi đã trình bày phương pháp phân loại văn bản dựa trên mô hình LSTM.Thì trong đó LSTM có các cổng giúp lọc thông tin như cổng đầu vào, cổng nhớ. Nhờ đó, các dữ liệu có ích từ thời điểm rất lâu trong quá khứ vẫn có tác dụng điều chỉnh. Với những yêu cầu và mục đích đã đặt ra cho hệ thống, bài toán bao gồm hai bước chính là đào tạo và dự đoán.

****

Hình 2.2.1: Sơ đồ cấu trúc hệ thống.

Như đã nói ở phần một thì mô hình LSTM bao gồm ba cổng là cổng ra, cổng quên và cổng dầu vào. Trong quá trính thực hiện các cổng được tính toán như sau:

Ở bước đầu tiên, tế bào LSTM quyết định những thông tin cần được loại bỏ từ các trạng thái tế bào bên trong ở bước thời gian trước đó S𝑡−1 . Giá trị 𝑓𝑡 của cổng quên tại bước thời gian t được tính toán dựa trên giá trị đầu vào hiện tại 𝑥 , giá trị đầu ra ℎ𝑡−1 từ tế bào LSTM ở bước trước đó và độ lệch (bias) 𝑏𝑓 của cổng quên. Hàm sigmoid biến đổi tất cả các giá trị kích hoạt (activation value) về miền giá trị trong khoảng từ 0 và 1 theo công thức:

𝑓𝑡 = (𝑊𝑓,𝑥 𝑥𝑡 + 𝑊𝑓,ℎ ℎ𝑡−1 + 𝑏𝑓)

Ở bước thứ 2, tế bào LSTM xác định những thông tin nào cần được thêm vào các trạng thái tế bào bên trong st . Bước này bao gồm hai quá trình tính toán đối với ŝ𝑡 và 𝑓 . ŝ𝑡 biểu diễn những thông tin có thể được thêm vào các trạng thái tế bào bên trong:

ŝ𝑡 = 𝑡𝑎𝑛ℎ(𝑊𝑠 ,𝑥 𝑥𝑡 + 𝑊ŝ ,ℎ ℎ𝑡−1 + 𝑏ŝ )

Giá trị 𝑖𝑡 của cổng đầu vào tại bước thời gian t được tính:

𝑖𝑡 = (𝑊𝑖,𝑥 𝑥𝑡 + 𝑊𝑖,ℎ ℎ𝑡−1 + 𝑏𝑖)

Ở bước tiếp theo, giá trị mới của trạng thái tế bào bên trong 𝑠𝑡 được tính toán dựa trên kết quả thu được từ các bước trên:

𝑠𝑡 = 𝑓𝑡 ∗ 𝑠𝑡−1 + 𝑖𝑡 ∗ ŝ𝑡

Cuối cùng, giá trị đầu ra ℎ :

𝑜𝑡 = (𝑊𝑜,𝑥 𝑥𝑡 + 𝑊𝑜,ℎ ℎ𝑡−1 + 𝑏𝑜)

ℎ𝑡 = 𝑜𝑡 ∗ 𝑡𝑎𝑛ℎ(𝑠𝑡)

Trong đó:

𝑊𝑠 ,𝑥 , 𝑊ŝ ,ℎ , 𝑊𝑓,𝑥 , 𝑊𝑓,ℎ , 𝑊𝑖,𝑥 , 𝑊𝑖,ℎ là các ma trận trọng số trong mỗi tế bào LSTM. 𝑏 , 𝑏𝑠 , 𝑏𝑖 , 𝑏𝑜 là các vector bias.

**Chương Trình Code của sản phẩm:**

**

Hình 2.2.2:Chương trình code.

**Thư viện sử dụng:**

• **nltk** : Bộ công cụ Ngôn ngữ Tự nhiên,là một nền tảng dẫn đầu để xây dựng các chương trình Python làm việc với dữ liệu ngôn ngữ của con người.

• **json** : để load các tệp json trực tiếp vào Python.

• **pickle** : để có thể đọc được file \*.pkl

• **numpy :** để có thể thực hiện các phép toán đại số tuyến tính.

• **keras** : là framework học sâu mà trong mô hình sẽ sử dụng

Demo sản phẩm

Chatbot sẽ tự động trả lời câu hỏi dựa trên một số mẫu câu hỏi mà Chatbot đã được người lập trình huấn luyện sẵn.



Hình 2.3.1:Giao diện chính của chương trình

# 

# Kết luận

* Kết quả đạt được:
* Về lý thuyết:

- Nắm được quy trình xây dựng một chương trình python.

- Nắm được những lý thuyết về mạng neural nhân tạo, cũng như cách xử lý dữ liệu và nguyên lý làm việc của các thư viện

-Nắm được loại mạng được Sử dụng để ứng dụng vào chatbot.

* Về thực hành:

- Xây dựng và huấn luyện thành công Chatbot với chức năng tương tác với một số câu hỏi của người sử dụng.

- Xây dựng được giao diện website tương tác với Chatbot đẹp mắt và dễ dàng tương tác.

* Hạn chế :
* Cơ sở dữ liệu chưa đủ lớn và chưa tối ưu.

# Tài liệu tham khảo

[1] https://stackoverflow.com/questions/24901637/what-is-a-recurrent-neural-network-what-is-a-long-short-term-memory-lstm-netw

[2] https://www.w3schools.com/ai/ai\_neural\_networks.asp

[3] https://www.codecademy.com/learn/paths/build-chatbots-with-python

[4] https://product.vinbigdata.org/tag/chatbot/