

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN VIỄN THÔNG



LUẬN VĂN KỸ SƯ

AI CHATBOT HỖ TRỢ THÔNG TIN
KÝ TÚC XÁ

SINH VIÊN THỰC HIỆN:
LÝ TRUNG KIÊN - 1611682

GIẢNG VIÊN HƯỚNG DẪN:
PGS. TS. HÀ HOÀNG KHA

TP. HỒ CHÍ MINH, 07/2021

LỜI CẢM ƠN

Dầu tiên, tôi xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô của Trường Đại học Bách Khoa tp.HCM, đặc biệt là đến thầy **Hà Hoàng Kha**, người đã nhiệt tình hướng dẫn, chỉ bảo tôi trong quá trình thực hiện và hoàn thành luận văn.

Tôi cũng xin cảm ơn gia đình, bạn bè, các anh chị đã giúp đỡ, ủng hộ tôi trong quá trình làm luận văn này.

Mặc dù đã cố gắng rất nhiều, nhưng luận văn không tránh khỏi những thiếu sót; tôi rất mong nhận được sự thông cảm, chỉ dẫn, giúp đỡ và đóng góp ý kiến của quý thầy cô, hội đồng đề cài thiện hơn cho việc xây dựng pháp triển đề tài sau này, cũng như lấy đó làm kinh nghiệm để ứng dụng cho công việc sau này.

Tôi xin trân trọng cảm ơn!

TPHCM, ngày 25 tháng 07 năm 2021
Sinh viên thực hiện

Lý Trung Kiên

LỜI CAM ĐOAN

Tôi tên: Lý Trung Kiên, là sinh viên chương trình PFIEV chuyên ngành Viễn thông khoá 2016 tại Đại học Quốc gia thành phố Hồ Chí Minh - Trường Đại học Bách Khoa.

Tôi xin cam đoan những nội dung sau đều là sự thật:

- Luận văn "AI Chatbot hỗ trợ thông tin Ký Túc Xá" hoàn toàn do tôi thực hiện.
- Các tài liệu trích dẫn trong luận văn được tham khảo từ các nguồn thực tế, có uy tín và độ chính xác cao.
- Các số liệu thống kê và kết quả của công trình này được tôi thực hiện một cách độc lập và trung thực.

TPHCM, ngày 25 tháng 07 năm 2021
Sinh viên thực hiện

Lý Trung Kiên

TÓM TẮT LUẬN VĂN

Hiện nay, việc sống trong môi trường đông dân cư như các chung cư, ký túc xá, toà nhà rất phổ biến, đặc biệt là tại các thành phố lớn. Và tất nhiên trong những môi trường đó đều có những nguyên tắc, nội quy chung để mọi người tuân theo, đảm bảo cuộc sống thoải mái cho tất cả mọi người ở trong đó. Tuy nhiên không phải ai cũng nắm rõ các quy định đó dẫn đến những khó khăn cho những người quản lý toà nhà, ảnh hưởng đến thoải mái của người khác. Dù công nghệ ngày càng phát triển, đặc biệt là sự xuất hiện những mạng xã hội, diễn đàn giúp chúng ta dễ dàng kết nối giao tiếp với nhau, những cũng chỉ giúp những nhà quản lý toà nhà được một phần thôi.

Nhận ra những khó khăn đó, đồng thời muốn áp dụng một số kiến thức học tập được để ứng dụng vào cuộc sống, em quyết định tiến hành đề tài "AI Chatbot hỗ trợ thông tin Ký Túc Xá". Với các ứng dụng thành tựu của trí tuệ nhân tạo đặc biệt là máy học, cũng như phương pháp xây dựng Web Server theo mô hình chủ khán, tích hợp kết nối với mạng xã hội Facebook, giúp cho đề tài mang tính linh hoạt, nhanh chóng và dễ tiếp cận hơn.

ABSTRACT

Currently, living in residential environments such as apartments, dormitories, and buildings is very common, especially in big cities. And all in those schools have common rules and regulations for everyone to follow, ensuring a comfortable life for everyone in them. However, not everyone is aware of these regulations to the difficulties for building managers, affecting the comfort of others. Although technology is increasingly developing, especially the emergence of social networks, forums that make it easy for them to connect with each other, only help building managers have a part.

Realizing those difficulties, and wanting to apply some knowledge files to apply in life, I decided on the topic "AI Chatbot supporting information for Dormitory". Applying artificial intelligence, especially machine learning, as well as methods of making Web Server according to other models, integrating connection with Facebook social network, making the topic flexible, fast, and more accessible.

Mục lục

TÓM TẮT LUẬN VĂN	iii
ABSTRACT	iv
DANH SÁCH HÌNH VẼ	viii
DANH SÁCH BẢNG	ix
Một số kí hiệu viết tắt	x
1 GIỚI THIỆU ĐỀ TÀI	1
1.1 Đặt vấn đề	1
1.2 Thông tin liên quan	2
1.3 Mục tiêu của luận văn	3
1.4 Nội dung luận văn	4
2 MẠNG THẦN KINH NHÂN TẠO	5
2.1 Cấu trúc mạng thần kinh nhân tạo	5
2.1.1 Cấu trúc tế bào mạng thần kinh nhân tạo	5
2.1.2 Mạng neural nhân tạo	7
2.1.3 Mô hình máy học	8
2.1.4 Hàm kích hoạt	9
2.2 Hàm mất mát và tối ưu hàm mất mát	14
2.2.1 Hàm mất mát	14
2.2.2 Giải thuật xuồng dốc (GD)	16
2.2.3 Giải thuật xuồng dốc ngẫu nhiên (SGD)	17
2.2.4 Hệ số học và hệ số đà	17
2.3 Hàm lan truyền ngược	18
2.4 Lỗi ước lượng tham số	19
2.4.1 Tổng quan	19
2.4.2 Độ lệch và phương sai	19

MỤC LỤC

2.4.3 Kỹ thuật Dropout	20
3 MÔ HÌNH MẠNG NEURAL CHO CHATBOT	22
3.1 Tổng quan về Chatbot	22
3.2 Xây dựng và huấn luyện mạng neural	23
3.2.1 Xây dựng huấn luyện	23
3.2.2 Thiết kế xây dựng và huấn luyện mạng neural	25
3.3 Xử lý đầu vào và phân luồng theo yêu cầu	28
4 HỆ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG MÁY CHỦ WEB	31
4.1 Cơ sở dữ liệu theo DBMS	31
4.1.1 Lược đồ cơ sở dữ liệu	31
4.1.2 Xây dựng cơ sở dữ liệu trong MySQL	32
4.1.3 Kết quả xây dựng bảng	35
4.2 Máy chủ web	35
4.2.1 Tổng quan	35
4.2.2 Flask Server	37
4.2.3 Gunicorn	37
4.2.4 Supervisor	40
4.2.5 Caddy	42
5 PHƯƠNG PHÁP GIAO TIẾP NGƯỜI DÙNG	45
5.1 Giao thức HTTPS	45
5.1.1 Một số khái niệm	45
5.1.2 Phương thức GET và POST	46
5.1.3 Khuông dạng thông điệp yêu cầu	48
5.1.4 Khuông dạng thông điệp trả lời	50
5.2 Giao tiếp người dùng thông qua giao diện web	52
5.3 Giao tiếp người dùng thông qua giao diện messenger facebook	54
6 KẾT QUẢ ĐỀ TÀI VÀ ĐỊNH HƯỚNG PHÁT TRIỂN	57
6.1 Kết quả đề tài	57
6.1.1 Đối với văn bản	58
6.1.2 Đối với âm thanh	65
6.2 Ưu điểm và hạn chế của đề tài	67
6.3 Định hướng phát triển đề tài	67
TÀI LIỆU KHAM KHẢO	68

Danh sách hình vẽ

1.1	Một số câu hỏi thường gặp	2
1.2	Ứng dụng chatbot trong hỗ trợ, thông tin bầu cử 2021	3
2.1	Các mô hình tế bào thần kinh sinh học và nhân tạo	6
2.2	Các tầng trong mạng neural	8
2.3	Một số hàm kích hoạt ở lớp ẩn	11
2.4	Cách chọn hàm kích hoạt cho các lớp ẩn trong mô hình mạng neural	12
2.5	Chương trình ví dụ bằng Python	13
2.6	Kết quả của ví dụ	13
2.7	Cách chọn hàm kích hoạt cho đầu ra trong mô hình mạng neural	14
2.8	Sự khác biệt giữa SGD và GD.	17
2.9	Sự đánh đổi giữa Bias và Variance	20
2.10	Kỹ thuật dropout	21
3.1	Hệ thống xử lý trong Chatbot	23
3.2	Cấu trúc file json.	24
3.3	Lựa chọn hàm mất mát.	25
3.4	Lựa chọn hệ số đà.	26
3.5	Lựa chọn hệ số học.	26
3.6	Lựa chọn hệ số học.	26
3.7	Kết quả huấn luyện các model	27
3.8	Mô hình phân luồng câu hỏi và tìm kiếm câu trả lời	28
3.9	Kết quả phương sai cho model switch	29
4.1	Sơ đồ quan hệ thực thể	32
4.2	Cú pháp kết nối ứng dụng với MySQL	34
4.3	Lược đồ cơ sở dữ liệu được xây dựng từ MySQL	35
4.4	Cấu hình máy chủ server.	36
4.5	Lực đồ liên hệ giữa các thành phần trong hệ thống máy chủ Web.	36

DANH SÁCH HÌNH VẼ

4.6	Cấu trúc hoạt động của WSGI	38
4.7	Cấu hình các quá trình của Gunicorn bằng Supervisor	42
4.8	Caddy	43
4.9	Cấu hình máy chủ Web bằng Caddyfile.	44
5.1	Thông điệp yêu cầu.	48
5.2	Thông điệp trả lời.	50
5.3	Quá trình xử lý giao tiếp bằng văn bản hoặc giọng nói	53
5.4	Giao tiếp chatbot	53
5.5	Quá trình xử lý ở server	54
5.6	Quá trình giao tiếp với server Facebook	54
5.7	Cài đặt Webhook.	56
5.8	Quá trình lấy và gửi câu trả lời âm thanh.	56
6.1	Kết quả giao diện trên website.	57
6.2	Hệ thống quá tải khi phải xử lý nhiều yêu cầu cùng lúc.	67

Danh sách bảng

2.1	Điểm chung giữa mô hình mạng lưới neural sinh học và nhân tạo.	6
2.2	So sánh giữa mô hình mạng lưới neural sinh học và nhân tạo.	7
2.3	Kiểu kết hợp hàm kích hoạt và hàm mất mát lớp cuối cùng.	15
6.1	Kết quả xử lý với đầu vào văn bản.	58
6.2	Kết quả xử lý với đầu vào giọng nói.	66

MỘT SỐ KÝ HIỆU VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
ANN	Artifical Neural Network	Mạng thần kinh nhân tạo
API	Application Programming Interface	Giao diện lập trình ứng dụng
CNN	Convolutional Neural Network	Mạng thần kinh tích chập
CSDL	Database	Cơ sở dữ liệu
DBMS	Database Management System	Hệ cơ sở dữ liệu
ER	Entity Relationship	Quan hệ thực thể
HTML	Hypertext Markup Language	Ngôn ngữ đánh dấu
HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
KTX	Bach Khoa Dormitory of Ho Chi Minh City University of Technology	Ký túc xá Đại học Bách Khoa
MLP	Multilayer Perceptron	Perceptron nhiều lớp
LSTM	Long Short Term Memory	Mạng bộ nhớ dài ngắn
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
ReLU	Rectified Linear Unit	Đơn vị tuyến tính chỉnh lưu
RNN	Recurrent Neural Network	Mạng thần kinh hồi quy
SGD	Stochastic Gradient Descent	Giải thuật xuống dốc ngẫu nhiên
WSGI	Web Server Gateway Interface	Giao diện cổng máy chủ web
URL	Uniform Resource Locator	Định vị tài nguyên thống nhất

Chương 1

GIỚI THIỆU ĐỀ TÀI

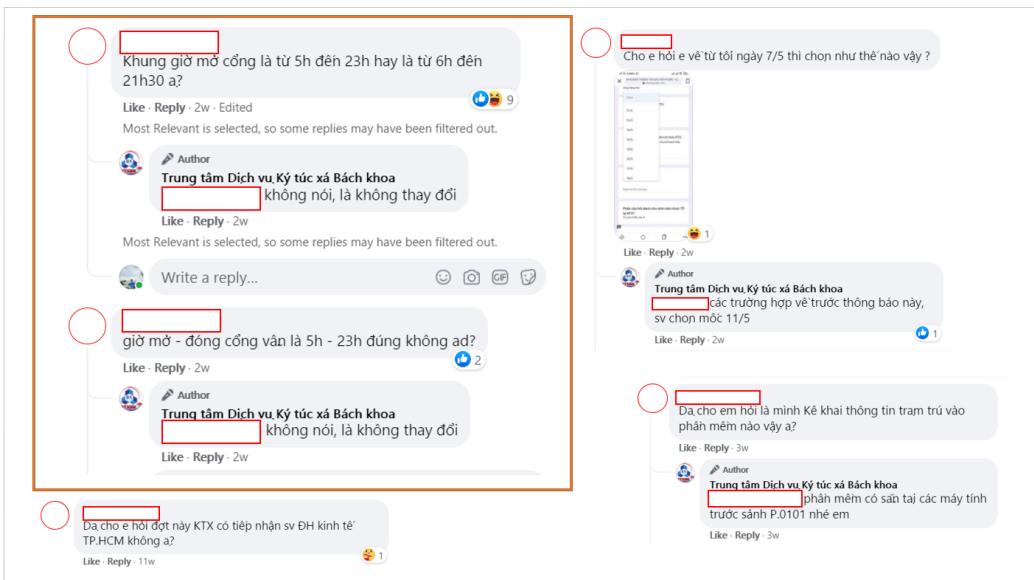
Chương này sẽ giới thiệu về ý tưởng về đề tài, mục tiêu và nhiệm vụ của đề tài, các phương pháp xây dựng hệ thống cũng như khái quát chung nhất về cấu trúc và hoạt động của hệ thống.

1.1 Đặt vấn đề

Sau một thời gian sinh sống ở Ký túc xá Bách Khoa (KTX), Trường Đại học Bách Khoa, em nhận thấy rằng sẽ có nhiều sự thắc mắc, băn khoăn khi có sự thay đổi, thông báo mới từ KTX như: quy định về thời gian giới nghiêm, nhân sự của KTX, sử dụng trang thiết bị hiện có, mới bổ xung,... Đặc biệt là từ các sinh viên mới vô hoặc những ai đang tìm hiểu để chuyển vào KTX.

Hiểu cho những khó khăn đó, em quyết định thực hiện đề tài: “AI Chatbot hỗ trợ thông tin trong Ký túc xá” với mong muốn giúp các bạn tiếp cận được các thông tin dễ dàng, nhanh chóng và chính xác. Hơn nữa cũng giảm phần nào gánh nặng công việc của các cán bộ KTX, đặc biệt là giải đáp những thắc mắc ở ngoài giờ làm việc của họ.

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI



Hình 1.1: Một số câu hỏi thường gặp

1.2 Thông tin liên quan

Cho dù chỉ mới phổ biến những năm gần đây nhưng ít ai biết rằng ý tưởng xây dựng chatbot đã xuất hiện từ những năm 50 về trước. Năm 1950, bài báo nổi tiếng của Alan Turing "Máy tính và trí thông minh" được xuất bản, đề xuất cái mà ngày nay được gọi là phép thử Turing như một tiêu chí của trí thông minh. Tiêu chí này phụ thuộc vào khả năng của một chương trình máy tính đóng giả con người trong một cuộc trò chuyện bằng văn bản trong thời gian thực với một thám phán con người đến mức mà thám phán không thể phân biệt một cách đáng tin cậy — chỉ dựa trên nội dung hội thoại — giữa chương trình và một con người thực sự. Sự nổi tiếng của bài kiểm tra do Turing đề xuất đã kích thích sự quan tâm lớn đến chương trình ELIZA của Joseph Weizenbaum, được công bố vào năm 1966, dường như có thể đánh lừa người dùng tin rằng họ đang trò chuyện với một con người thật.

Trong những năm đầu của thế kỷ XXI, đã có những đột phá trong các lĩnh vực trí tuệ nhân tạo. Các chatbot được ứng dụng các công nghệ này, hoạt động dựa trên thuật toán và nền tảng đám mây, cho phép chúng thích ứng và "học" dựa trên tương tác của chúng với con người. Vào năm 2016, Facebook Messenger đã cho phép các nhà phát triển đặt chatbot trên nền tảng của họ. Đã có 30.000 bot được tạo cho Messenger trong sáu tháng đầu tiên, tăng lên 100.000 vào tháng 9 năm 2017. Nhiều ngân hàng, công ty bảo hiểm, công ty truyền thông, công ty thương mại điện tử, hàng hàng không, chuỗi khách

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

sạn, nhà bán lẻ, nhà cung cấp dịch vụ chăm sóc sức khỏe, tổ chức chính phủ và chuỗi nhà hàng đã sử dụng chatbot để trả lời các câu hỏi đơn giản, tăng mức độ tương tác của khách hàng, để quảng cáo và cung cấp các cách bổ sung để đặt hàng từ chúng. Một nghiên cứu năm 2017 cho thấy 4% công ty đã sử dụng chatbots. Theo một nghiên cứu năm 2016, 80% doanh nghiệp cho biết họ dự định có một công ty vào năm 2020.

Tại Việt Nam, thời gian gần đây, Chatbot đã được sử dụng rộng rãi trong các lĩnh vực kinh doanh, tài chính,... Mới đây nhất, chatbot đã lần đầu tiên được sử dụng để giải đáp những thắc mắc về bầu cử trong đợt bầu cử năm 2021.



Hình 1.2: Ứng dụng chatbot trong hỗ trợ, thông tin bầu cử 2021

Hiện nay chatbot được ứng dụng khá rộng rãi trong nhiều lĩnh vực đời sống. Tuy nhiên việc triển khai chatbot trong các khu vực có mật độ dân cư lớn như các tòa nhà, KTX vẫn còn khá mới mẻ. Một số lý do nó bị hoen chế do tính thiếu linh hoạt cũng như công việc phản hồi lập tức chưa được xem trọng. Do đó chatbot chỉ hỗ trợ tư vấn phần nào những vấn đề cơ bản và mang tính khuôn mẫu, giúp giảm tải phần nào áp lực của các nhân viên tư vấn, chứ không hoàn toàn thay thế vai trò của họ.

1.3 Mục tiêu của luận văn

Trong quá trình nghiên cứu và hiện thực mô hình chatbot, em xác định được một số vấn đề về phương pháp xây dựng các thành phần của hệ thống như sau:

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

- Sử dụng phương pháp máy học để đem lại hiệu quả cao và chính xác hơn cho mô hình. Lựa chọn dạng Model và các thông số mạng neural phù hợp.
- Cơ sở dữ liệu phù hợp, cấu trúc truy vấn và đảm bảo tính nhất quán, toàn vẹn dữ liệu, an toàn thông tin.
- Xây dựng các phương pháp giao tiếp với người dùng thông qua văn bản, âm thanh thông qua website, messenger facebook.

1.4 Nội dung luận văn

- Xây dựng cho chatbot mạng neural cơ bản theo mô hình Retrieval-Base.
- Xây dựng phương pháp lưu trữ và tổ chức các hệ cơ sở dữ liệu.
- Xây dựng những thuật toán truy suất thông tin phù hợp với yêu cầu người dùng.
- Xây dựng các phương pháp giao tiếp với người dùng bằng văn bản, âm thanh thông qua website, messenger facebook.

Cấu trúc của bài báo cáo luận văn này như sau:

- **Chương 1:** Giới thiệu tổng quát đề tài và các phương pháp được lựa chọn.
- **Chương 2:** Trình bày các vấn đề cơ bản của mạng thần kinh nhân tạo.
- **Chương 3:** Xây dựng và huấn luyện mạng thần kinh nhân tạo cho hệ thống chatbot.
- **Chương 4:** Trình bày cách thức xây dựng hệ cơ sở dữ liệu và hệ thống máy chủ web.
- **Chương 5:** Xây dựng mô hình máy chủ và phương pháp giao tiếp với người dùng thông qua 2 công cụ là: website và messenger facebook.
- **Chương 6:** Báo cáo kết quả thực hiện được, từ đó rút ra kinh nghiệm và định hướng phát triển đề tài trong tương lai.

Chương này đã giới thiệu về đề tài cũng như trình bày mục đích, cấu trúc của luận văn. Chương sau sẽ trình bày 1 số kiến thức mà sẽ được áp dụng trong đề tài này.

Chương 2

MẠNG THẦN KINH NHÂN TẠO

Ở chương này, ta sẽ đề cập đến các vấn đề cơ bản của mạng neural từ cấu tạo của mạng từ một tế bào neural nhân tạo cho đến cả một mạng neural nhân tạo; các kỹ thuật huấn luyện và tối ưu như kỹ thuật lan truyền ngược, kỹ thuật tối ưu hàm mất mát và kỹ thuật hạn chế lỗi ước lượng tham số,... Từ đó, ta đưa ra các lựa chọn về phương pháp xây dựng mạng neural để xử lý cho mô hình chatbot.

2.1 Cấu trúc mạng thần kinh nhân tạo

2.1.1 Cấu trúc tế bào mạng thần kinh nhân tạo

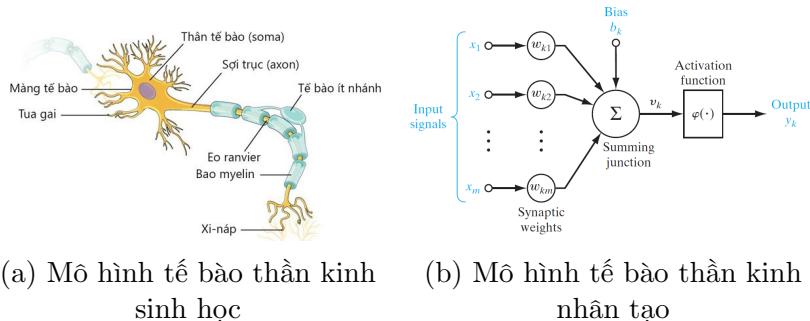
Dựa vào cấu trúc và chức năng của tế bào thần kinh sinh học (BNN), người ta đã tạo ra mô hình tế bào thần kinh nhân tạo (ANN). Và chúng có những điểm tương đồng nhất định:

CHƯƠNG 2. MẠNG THẦN KINH NHÂN TẠO

BNN	ANN
Thân tế bào	Nút
Tua gai	Đầu vào
Synapse	Tập khớp thần kinh: Mỗi khớp chứa trọng số (weight). Đầu vào x_i cho một khớp j của mảnh neural k nhân với trọng số w_{kj} .
Sợi trực	Hàm kích hoạt: giới hạn giá trị đầu ra của neural trong một dãy giá trị nào đó, thông số bias b_k như mức ngưỡng quyết định đầu ra.

Bảng 2.1: Điểm chung giữa mô hình mạng lưới neural sinh học và nhân tạo.

Tuy nhiên vẫn có sự khác nhau cơ bản ở một số tiêu chí.



Hình 2.1: Các mô hình tế bào thần kinh sinh học và nhân tạo

CHƯƠNG 2. MẠNG THẦN KINH NHÂN TẠO

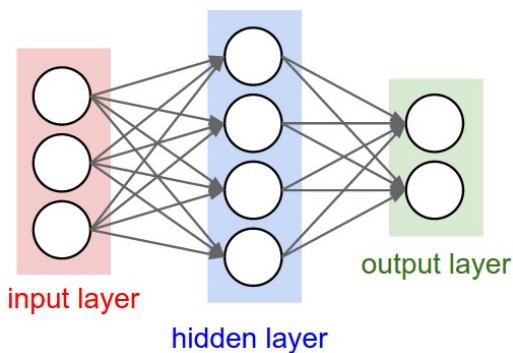
Tiêu chí	BNN	ANN
Cơ chế	Song song ô ạt, chậm nhưng vượt trội hơn ANN	Song song ô ạt, nhanh nhưng kém hơn BNN
Kích thước	10^{11} tế bào thần kinh và 10^{15} liên kết với nhau	10^2 đến 10^4 nút chủ yếu phụ thuộc vào loại ứng dụng và nhà thiết kế mạng
Học tập	Họ có thể chịu đựng sự mơ hồ	Dữ liệu có cấu trúc và định dạng rất chính xác được yêu cầu để chống lại sự mơ hồ
Khả năng chịu lỗi	Hiệu suất suy giảm với thậm chí hư hỏng một phần	Nó có khả năng hoạt động mạnh mẽ, do đó có khả năng chịu lỗi
Khả năng lưu trữ	Lưu trữ thông tin trong khớp thần kinh	Lưu trữ thông tin trong các vị trí bộ nhớ liên tục

Bảng 2.2: So sánh giữa mô hình mạng lưới neural sinh học và nhân tạo.

2.1.2 Mạng neural nhân tạo

Mạng neural nhân tạo (ANN) là một hệ thống máy tính hiệu quả có chủ đề trung tâm được vay mượn từ sự tương tự của mạng nơ-ron sinh học. ANN cũng được đặt tên là “hệ thống thần kinh nhân tạo” hoặc “hệ thống xử lý phân tán song song” hoặc “hệ thống kết nối”. ANN có được một tập hợp lớn các đơn vị được kết nối với nhau theo một số mẫu để cho phép giao tiếp giữa các đơn vị. Những đơn vị này, còn được gọi là nút hoặc nơ-ron, là những bộ xử lý đơn giản hoạt động song song.

Mạng neural là sự kết hợp của các tầng perceptron hay còn được gọi là perceptron đa tầng (multilayer perceptron) như hình vẽ bên dưới:



Hình 2.2: Các tầng trong mạng neural

Một mạng neural sẽ có 3 kiểu tầng:

- **Tầng vào** (input layer): Là tầng bên trái cùng của mạng thể hiện cho các đầu vào của mạng.
- **Tầng ra** (output layer): Là tầng bên phải cùng của mạng thể hiện cho các đầu ra của mạng.
- **Tầng ẩn** (hidden layer): Là tầng nằm giữa tầng vào và tầng ra thể hiện cho việc suy luận logic của mạng.

Lưu ý rằng, một mạng neural chỉ có 1 tầng vào và 1 tầng ra nhưng có thể có nhiều tầng ẩn.

Ở mỗi tầng, số lượng các nút mạng (nơ-ron) có thể khác nhau tuỳ thuộc vào bài toán và cách giải quyết. Nhưng thường khi làm việc người ta để các tầng ẩn có số lượng nơ-ron bằng nhau. Ngoài ra, các nơ-ron ở các tầng thường được liên kết đôi một với nhau tạo thành mạng kết nối đầy đủ (full-connected network). Khi đó ta có thể tính được kích cỡ của mạng dựa vào số tầng và số nơ-ron. Ví dụ ở hình trên ta có:

- 4 tầng mạng, trong đó có 22 tầng ẩn
- $3 + 4 * 2 + 1 = 12$ nút mạng
- $3 * 4 + 4 * 4 + 4 * 1 + (4 + 4 + 1) = 41$ tham số

2.1.3 Mô hình máy học

Ở đề tài này, ta sử dụng mô hình học có giám sát nên chúng ta hãy tìm hiểu qua về mô hình này.

CHƯƠNG 2. MẠNG THẦN KINH NHÂN TẠO

Phần lớn học máy thực tế sử dụng phương pháp học có giám sát. Học có giám sát là nơi bạn có các biến đầu vào (x) và một biến đầu ra (Y) và bạn sử dụng một thuật toán để học hàm ánh xạ từ đầu vào đến đầu ra.

$$Y = f(X) \quad (2.1)$$

Mục đích là làm gần đúng hàm ánh xạ sao cho khi bạn có dữ liệu đầu vào mới (x), bạn có thể dự đoán các biến đầu ra (Y) cho dữ liệu đó. Nó được gọi là học có giám sát vì quá trình học thuật toán từ tập dữ liệu đào tạo có thể được coi như một "giám sát viên" của quá trình học tập. Thuật toán lặp đi lặp lại các dự đoán trên dữ liệu huấn luyện và được sửa bởi "giám sát viên". Việc học tập dừng lại khi thuật toán đạt được mức hiệu suất có thể chấp nhận được.

Các vấn đề học tập có giám sát có thể được nhóm lại thành các bài toán hồi quy và phân loại.

- **Phân loại:** Một vấn đề phân loại là khi biến đầu ra là một danh mục, chẳng hạn như "đỏ" hoặc "xanh" hoặc "bệnh" và "không bệnh".
- **Hồi quy:** Một vấn đề hồi quy là khi biến đầu ra là một giá trị thực, chẳng hạn như "đô la" hoặc "trọng lượng".

Một số dạng vấn đề phổ biến được xây dựng dựa trên phân loại và hồi quy bao gồm đề xuất và dự đoán chuỗi thời gian tương ứng. Một số ví dụ phổ biến về thuật toán học máy được giám sát là:

- Hồi quy tuyến tính cho các bài toán hồi quy.
- Thuật toán rừng ngẫu nhiên (random forest) cho các bài toán phân loại và hồi quy.
- Hỗ trợ máy vectơ (vector machines) cho các bài toán phân loại.

2.1.4 Hàm kích hoạt

Trong phần này, ta tìm hiểu một số hàm kích hoạt kích hoạt thông dụng để từ đó chọn ra hàm kích hoạt đầu vào phù hợp ở các lớp trong mạng neural. Một hàm kích hoạt trong mạng nơ-ron xác định cách tổng trọng số của đầu vào được chuyển thành đầu ra từ một nút hoặc các nút trong một lớp của mạng. Một mạng có thể có ba loại lớp: lớp đầu vào nhận đầu vào thô từ bên ngoài, lớp ẩn nhận đầu vào từ lớp đầu vào và chuyển đầu ra cho lớp đầu ra và lớp đầu ra đưa ra dự đoán.

Chúng ta hãy lần lượt xem xét các chức năng kích hoạt được sử dụng cho từng loại lớp.

- **Kích hoạt cho các lớp ẩn:** Một mạng nơ-ron có thể không có hoặc nhiều lớp ẩn. Có ba hàm kích hoạt phổ biến mà ta cần xem xét để sử dụng trong các lớp ẩn.
 - **Hàm Sigmoid:** còn được gọi là hàm logistic. Đây là chức năng tương tự được sử dụng trong thuật toán phân loại hồi quy logistic. Hàm nhận bất kỳ giá trị thực nào làm giá trị đầu vào và đầu ra trong phạm vi từ 0 đến 1. Đầu vào càng lớn (càng dương) thì giá trị đầu ra càng gần với 1,0, trong khi đầu vào càng nhỏ (càng âm) thì giá trị càng gần đầu ra sẽ là 0,0. Hàm kích hoạt sigmoid được tính như sau:

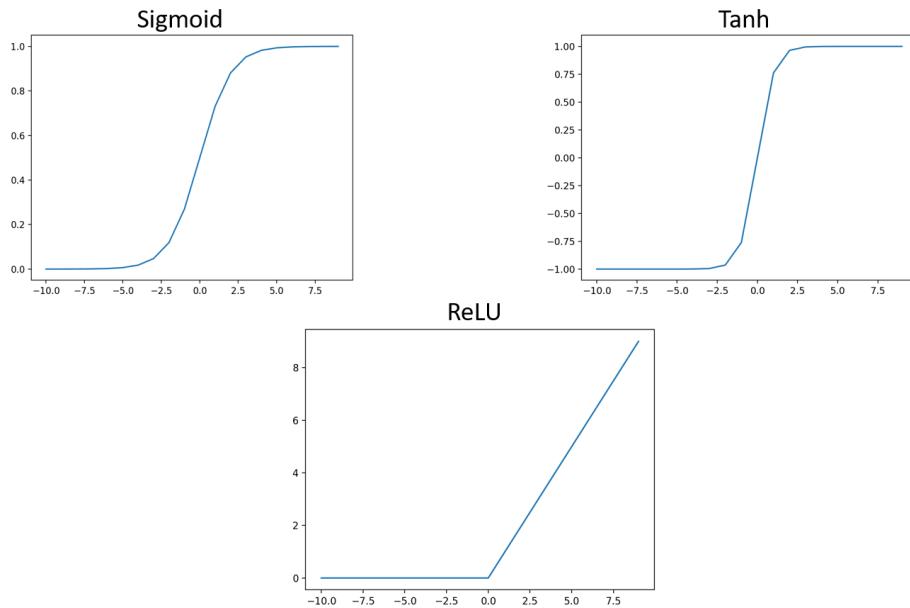
$$\varphi(v) = \frac{1}{1 + e^{-av}} \Rightarrow \varphi(v)' = a\varphi(v)(1 - \varphi(v)) \quad (2.2)$$

- **Hàm Tanh:** là hàm kích hoạt tiếp tuyến hyperbol còn được gọi đơn giản là hàm Tanh (cũng là "tanh" và "Tanh"). Nó rất giống với chức năng kích hoạt sigmoid và thậm chí có hình chữ S. Hàm nhận bất kỳ giá trị thực nào làm giá trị đầu vào và đầu ra trong phạm vi -1 đến 1. Đầu vào càng lớn (càng dương) thì giá trị đầu ra càng gần với 1,0, trong khi đầu vào càng nhỏ (càng âm) thì càng gần đầu ra sẽ là -1,0. Chức năng kích hoạt Tanh được tính như sau:

$$\varphi(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}} \Rightarrow \varphi(v)' = 1 - \varphi^2(v) \quad (2.3)$$

- **Hàm ReLU:** có lẽ là chức năng phổ biến nhất được sử dụng cho các lớp ẩn. Nó phổ biến vì nó vừa đơn giản để thực hiện vừa có hiệu quả khắc phục những hạn chế của các hàm kích hoạt phổ biến khác trước đây, chẳng hạn như Sigmoid và Tanh. Cụ thể, nó ít bị ảnh hưởng bởi các độ dốc biến mất ngăn cản việc đào tạo các mô hình sâu, mặc dù nó có thể gặp phải các vấn đề khác như các đơn vị bão hòa hoặc "chết". Hàm ReLU được tính như sau:

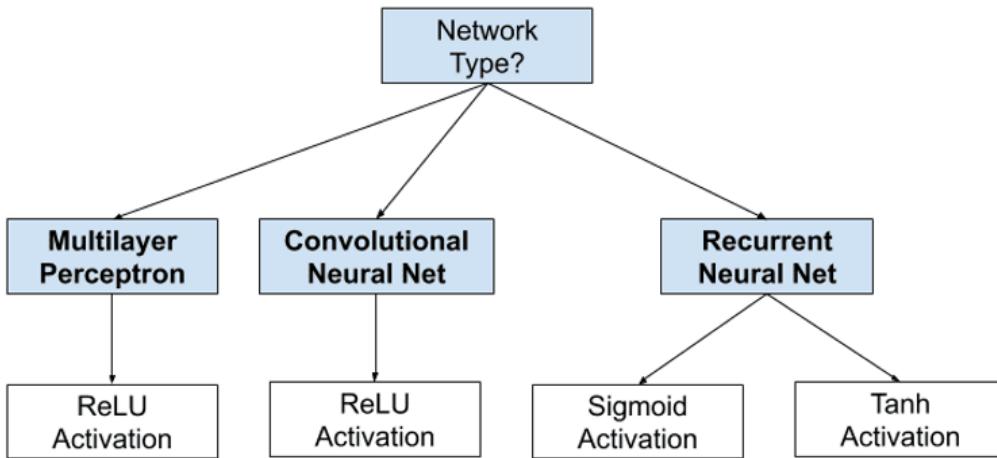
$$\varphi(v) = \begin{cases} 0 & (v \leq 0) \\ v & (v > 0) \end{cases} \Rightarrow \varphi(v)' = \begin{cases} 0 & (v \leq 0) \\ 1 & (v > 0) \end{cases} \quad (2.4)$$



Hình 2.3: Một số hàm kích hoạt ở lớp ẩn

Dựa vào những ưu điểm riêng của mỗi hàm mà ta lựa chọn ứng dụng trong mạng neural phù hợp, thông thường sẽ là:

- Perceptron nhiều lớp (MLP) : Hàm kích hoạt ReLU.
- Mạng thần kinh tích chập (CNN) : Hàm kích hoạt ReLU.
- Mạng thần kinh hồi quy (RNN) : Hàm kích hoạt Tanh và / hoặc Sigmoid.



Hình 2.4: Cách chọn hàm kích hoạt cho các lớp ẩn trong mô hình mạng neural

- **Hàm kích hoạt cho lớp đầu ra:**

- **Hàm kích hoạt đầu ra tuyến tính:**

Hàm kích hoạt tuyến tính còn được gọi là “nhận dạng” (nhận với 1,0) hoặc “không kích hoạt”. Điều này là do hàm kích hoạt tuyến tính không thay đổi tổng trọng số của đầu vào theo bất kỳ cách nào và thay vào đó trả về giá trị trực tiếp.

- **Hàm kích hoạt đầu ra Sigmoid:**

Hàm kích hoạt logistic đáng kể đã được mô tả trong phần trước. Các nhãn mục tiêu được sử dụng để huấn luyện một mô hình có hàm kích hoạt sigmoid trong lớp đầu ra sẽ có các giá trị 0 hoặc 1.

- **Hàm kích hoạt đầu ra Softmax:**

Hàm softmax xuất ra một vectơ có giá trị tổng bằng 1,0 có thể được hiểu là xác suất của thành viên lớp. Nó liên quan đến hàm argmax xuất ra giá trị 0 cho tất cả các tùy chọn và 1 cho tùy chọn đã chọn. Như vậy, đầu vào của hàm là một vectơ có giá trị thực và đầu ra là một vectơ có cùng độ dài với các giá trị có tổng bằng 1,0 như xác suất. Hàm softmax được tính như sau:

$$\varphi_i(v) = \frac{e^{v_i}}{\sum_{i=1}^n e^{v_i}} \quad (2.5)$$

Chúng ta không thể vẽ biểu đồ của hàm softmax, nhưng chúng ta có thể đưa ra một ví dụ về tính toán nó bằng Python.

CHƯƠNG 2. MẠNG THẦN KINH NHÂN TẠO

```
1 from numpy import exp
2
3 # softmax activation function
4 def softmax(x):
5     return exp(x) / exp(x).sum()
6
7 # define input data
8 inputs = [1.0, 3.0, 2.0]
9 # calculate outputs
10 outputs = softmax(inputs)
11 # report the probabilities
12 print(outputs)
13 # report the sum of the probabilities
14 print(outputs.sum())
```

Hình 2.5: Chương trình ví dụ bằng Python

Chạy ví dụ sẽ tính toán đầu ra softmax cho vector đầu vào. Sau đó, tổng các đầu ra của softmax đã được xác nhận thực sự bằng giá trị 1,0.

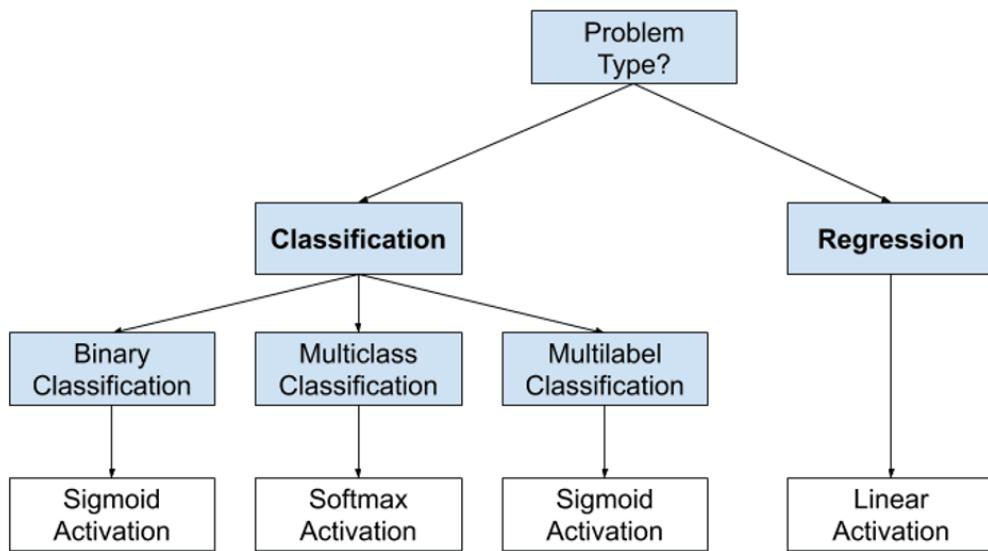
```
1 [0.09003057 0.66524096 0.24472847]
2 1.0
```

Hình 2.6: Kết quả của ví dụ

Hàm Softmax được sử dụng trong các mô hình nhiều lớp trong đó nó trả về xác suất của mỗi lớp, với lớp mục tiêu có xác suất cao nhất. Hàm Softmax chủ yếu xuất hiện trong hầu hết các lớp đầu ra của kiến trúc học sâu. Sự khác biệt chính giữa Sigmoid và Softmax AF là Sigmoid được sử dụng trong phân loại nhị phân trong khi Softmax được sử dụng cho các nhiệm vụ phân loại đa biến.

Chọn hàm kích hoạt cho lớp đầu ra của mình dựa trên loại vấn đề dự đoán mà bạn đang giải quyết và thông thường người ta lựa chọn như sau:

- Phân loại nhị phân : Một nút, kích hoạt sigmoid.
- Phân loại đa lớp : Một nút cho mỗi lớp, kích hoạt softmax.
- Phân loại đa nhãn : Một nút cho mỗi lớp, kích hoạt sigmoid.



Hình 2.7: Cách chọn hàm kích hoạt cho đầu ra trong mô hình mạng neural

Dựa vào các kết quả phân tích trên, ta sẽ lựa chọn hàm kích hoạt cho mạng neural ở lớp đầu vào và các lớp ẩn là hàm ReLU và ở lớp đầu ra là hàm Softmax.

2.2 Hàm mất mát và tối ưu hàm mất mát

2.2.1 Hàm mất mát

Hàm Loss là một phương pháp đánh giá “thuật toán của bạn mô hình hóa tập dữ liệu của bạn tốt như thế nào”. Nếu dự đoán của bạn bị tắt hoàn toàn, hàm mất mát của bạn sẽ xuất ra một con số cao hơn. Nếu chúng khá tốt, nó sẽ xuất ra một con số thấp hơn. Khi bạn điều chỉnh thuật toán của mình để thử và cải thiện mô hình của mình, hàm mất mát sẽ cho bạn biết liệu bạn có đang cải thiện hay không. ‘Mất mát’ giúp ta hiểu giá trị dự đoán khác với giá trị thực tế bao nhiêu. Các loại hàm mất mát:

- **Hàm mất hồi quy:** Mô hình hồi quy đề cập đến việc dự đoán một giá trị liên tục, ví dụ như diện tích sàn, số lượng phòng, kích thước của phòng, dự đoán giá phòng. Hàm mất mát được sử dụng trong bài toán hồi quy được gọi là "Hàm mất hồi quy".
- **Hàm mất phân loại nhị phân:** Phân loại nhị phân là một thuật toán dự đoán trong đó đầu ra có thể là một trong hai mục, được biểu

CHƯƠNG 2. MẠNG THẦN KINH NHÂN TẠO

thì bằng 0 hoặc 1. Đầu ra của thuật toán phân loại nhị phân là điểm dự đoán (chủ yếu). Vì vậy, việc phân loại xảy ra dựa trên ngưỡng giá trị (giá trị mặc định là 0,5). Nếu điểm dự đoán $>$ ngưỡng thì 1 khác 0.

- **Hàm Mất phân loại nhiều lớp:** Phân loại Đa Lớp là những vấn đề mô hình dự đoán trong đó có nhiều biến/ lớp mục tiêu hơn. Nó chỉ là phần mở rộng của vấn đề phân loại nhị phân.

Dựa vào ứng dụng khác nhau của cặp hàm kích hoạt và hàm mất mát lớp cuối cùng, kết hợp cho các nhiệm vụ khác nhau.

Loại vấn đề	Hàm kích hoạt lớp cuối	Hàm mất mát	Ví dụ
Phân loại nhị phân	sigmoid	binary_crossentropy	Phân loại chó vs mèo, Phân tích thành phần (pos/neg)
Phân loại nhiều lớp, một nhãn	softmax	categorical_crossentropy	MNIST có 10 lớp nhãn đơn (một dự đoán là một chữ số)
Phân loại nhiều lớp, nhiều nhãn	sigmoid	binary_crossentropy	Phân loại thẻ tin tức, một blog có thể có nhiều thẻ
Hồi quy về các giá trị tùy ý	Không có	mse	Dự đoán giá nhà (số nguyên/dấu phẩy động)
Hồi quy về các giá trị từ 0 đến 1	sigmoid	mse hoặc binary_crossentropy	Dánh giá tình trạng động cơ trong đó 0 bị hỏng, 1 là mới

Bảng 2.3: Kiểu kết hợp hàm kích hoạt và hàm mất mát lớp cuối cùng.

Trong thực tế, dữ liệu đầu vào và số chiều điều lớn, nên hàm mất mát trở nên khá phức tạp và ta khó có thể tìm được giá trị tối ưu bằng các công thức thông thường. Vì vậy, ta cần những phương pháp có thể đi đến giá trị tối ưu của hàm mất mát, đó là những phương pháp GD, SGD, Mini-batch GD,... Trong đề tài này, ta sẽ sử dụng phương pháp SGD và NAG (là một phương pháp mở rộng của GD).

2.2.2 Giải thuật xuống dốc (GD)

Gradient Descent (GD) là một thuật toán tối ưu lặp (iterative optimization algorithm) được sử dụng trong các bài toán Machine Learning và Deep Learning (thường là các bài toán tối ưu lồi — Convex Optimization) với mục tiêu là tìm một tập các biến nội tại (internal parameters) cho việc tối ưu models. Trong đó:

- Gradient: là tỷ lệ độ nghiêng của đường dốc (rate of inclination or declination of a slope). Về mặt toán học, Gradient của một hàm số là đạo hàm của hàm số đó tương ứng với mỗi biến của hàm. Đối với hàm số đơn biến, chúng ta sử dụng khái niệm Derivative thay cho Gradient.
- Descent: là từ viết tắt của descending, nghĩa là giảm dần.

Gradient Descent có nhiều dạng khác nhau như Stochastic Gradient Descent (SGD), Mini-batch SDG. Nhưng về cơ bản thì đều được thực thi như sau:

1. Khởi tạo biến nội tại.
2. Dánh giá model dựa vào biến nội tại và hàm mất mát (Loss function).
3. Cập nhật các biến nội tại theo hướng tối ưu hàm mất mát (finding optimal points).
4. Lặp lại bước 2, 3 cho tới khi thỏa điều kiện dừng.

Công thức cập nhật cho GD có thể được viết là:

$$\theta \leftarrow \theta - \eta \nabla_{\theta} f(\theta) \quad (2.6)$$

trong đó θ là tập các biến cần cập nhật, η là tốc độ học (learning rate), $\nabla_{\theta} f(\theta)$ là Gradient của hàm mất mát f theo tập θ .

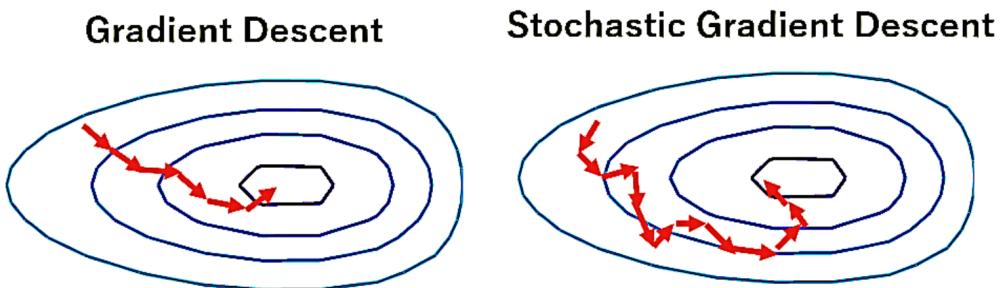
Điều kiện dừng của GD có thể là:

- Kết thúc tất cả các epochs đã được định sẵn.
- Giá trị của hàm mất mát đủ nhỏ và độ chính xác của model đủ lớn.
- Hàm mất mát có giá trị không thay đổi sau một số lần hữu hạn epochs.

2.2.3 Giải thuật xuống dốc ngẫu nhiên (SGD)

Trong thuật toán này, tại 1 thời điểm, ta chỉ tính đạo hàm của hàm mất mát dựa trên chỉ một điểm dữ liệu \mathbf{x}_i rồi cập nhật θ dựa trên đạo hàm này. Việc này được thực hiện với từng điểm trên toàn bộ dữ liệu, sau đó lặp lại quá trình trên. Thuật toán rất đơn giản này trên thực tế lại làm việc rất hiệu quả.

Mỗi lần duyệt một lượt qua tất cả các điểm trên toàn bộ dữ liệu được gọi là một epoch. Với GD thông thường thì mỗi epoch ứng với 1 lần cập nhật θ , với SGD thì mỗi epoch ứng với N lần cập nhật θ với N là số điểm dữ liệu. Nhìn vào một mặt, việc cập nhật từng điểm một như thế này có thể làm giảm đi tốc độ thực hiện 1 epoch. Nhưng nhìn vào một mặt khác, SGD chỉ yêu cầu một lượng epoch rất nhỏ (thường là 10 cho lần đầu tiên, sau đó khi có dữ liệu mới thì chỉ cần chạy dưới một epoch là đã có nghiệm tốt). Vì vậy SGD phù hợp với các bài toán có lượng cơ sở dữ liệu lớn và các bài toán yêu cầu mô hình thay đổi liên tục, tức online learning.



Hình 2.8: Sự khác biệt giữa SGD và GD.

Một điểm cần lưu ý đó là: sau mỗi epoch, chúng ta cần shuffle (xáo trộn) thứ tự của các dữ liệu để đảm bảo tính ngẫu nhiên. Việc này cũng ảnh hưởng tới hiệu năng của SGD.

Một cách toán học, quy tắc cập nhật của SGD là:

$$\theta = \theta - \eta \nabla_{\theta} J(\theta; \mathbf{x}_i; \mathbf{y}_i)$$

trong đó $J(\theta; \mathbf{x}_i; \mathbf{y}_i)$ là hàm mất mát với chỉ 1 cặp điểm dữ liệu (input, label) là $(\mathbf{x}_i, \mathbf{y}_i)$.

2.2.4 Hệ số học và hệ số đà

Hệ số học và hệ số đà là hai tham số rất quan trọng kiểm soát mức độ hiệu quả của thuật toán lan truyền ngược (sẽ được nhắc tới ở phần sau) đào

tạo mạng neural. Hệ số học là một hằng số dương kiểm soát tốc độ điều chỉnh các hệ số trọng lượng mới dựa trên thuật ngữ hiệu chỉnh gradient-descent được tính toán. Hệ số đà là một trọng lượng phụ được thêm vào các yếu tố trọng lượng làm tăng tốc độ điều chỉnh các yếu tố "đà". Hệ số đà giúp di chuyển quy trình tối thiểu hóa ra khỏi cực tiểu cục bộ.

2.3 Hàm lan truyền ngược

Để tính đạo hàm của hàm lỗi $\nabla J(\mathbb{W})$ trong mạng neural, ta sử dụng một giải thuật đặc biệt là giải thuật lan truyền ngược (backpropagation). Nhờ có giải thuật được sáng tạo vào năm 1986 này mà mạng neural được thực thi hiệu quả và được ứng dụng ngày càng nhiều cho tới tận ngày nay.

Về cơ bản phương pháp này được dựa theo quy tắc chuỗi đạo hàm của hàm hợp và phép tính ngược đạo hàm để thu được đạo hàm theo tất cả các tham số cùng lúc chỉ với 2 lần duyệt mạng. Giải thuật lan truyền ngược được thực hiện như sau:

1. Lan truyền tiến:

Lần lượt tính các $\mathbf{a}^{(l)}$ từ $l = 2 \rightarrow L$ theo công thức:

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \cdot \mathbf{a}^{(l-1)} \quad (2.7)$$

$$\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)}) \quad (2.8)$$

Trong đó, tầng vào $\mathbf{a}^{(1)}$ chính bằng giá trị vào của mạng \mathbf{x} .

2. Tính đạo hàm theo z ở tầng ra:

$$\frac{\partial J}{\partial z^{(l)}} = \frac{\partial J}{\partial a^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \quad (2.9)$$

với $\mathbf{a}^{(L)}$, $\mathbf{z}^{(L)}$ vừa tính được ở bước 1.

3. Lan truyền ngược:

Tính đạo hàm theo z ngược lại từ $l = (L - 1) \rightarrow 2$ theo công thức:

$$\frac{\partial J}{\partial z^{(l)}} = \frac{\partial J}{\partial z^{(l+1)}} \frac{\partial a^{(l)}}{\partial z^{(l)}} = \left((W^{(l+1)})^T \frac{\partial J}{\partial z^{(l+1)}} \right) \frac{\partial a^{(l)}}{\partial z^{(l)}} \quad (2.10)$$

với $\mathbf{z}^{(l)}$ tính được ở bước 1 và $\frac{\partial J}{\partial \mathbf{z}^{(l+1)}}$ tính được ở ngay vòng lặp trước.

4. **Tính đạo hàm:** Tính đạo hàm theo tham số ω bằng công thức:

$$\frac{\partial J}{\partial W^{(l)}} = \frac{\partial J}{\partial z^{(l)}} \frac{\partial z^{(l)}}{\partial W^{(l)}} = \frac{\partial J}{\partial z^{(l)}} (a^{l-1})^T \quad (2.11)$$

với $a^{(l-1)}$ tính được ở bước 1 và $\frac{\partial J}{\partial z^{(l)}}$ tính được ở bước 3.

2.4 Lỗi ước lượng tham số

2.4.1 Tổng quan

Mô hình của ta sau khi huấn luyện có thể đạt hiệu quả không tốt khi dự đoán với một dữ liệu mới. Chuyện này xảy ra là do mô hình của ta chưa tổng quát hoá được với toàn bộ tập dữ liệu. Nguyên nhân cũng khá dễ hiểu khi mà tập huấn luyện của ta chỉ là một tập nhỏ chưa thể đại diện cho toàn thể dữ liệu được và hơn nữa có thể nó còn bị nhiễu nữa. Người ta chia nguyên nhân ra làm 2 loại chính là **chưa khớp** (**Underfitting**) hoặc **quá khớp** (**Overfitting**) .

- Mô hình được coi là **chưa khớp** nếu nó chưa được chưa phù hợp với tập dữ liệu huấn luyện và cả các mẫu mới khi dự đoán. Nguyên nhân có thể là do mô hình chưa đủ độ phức tạp cần thiết để bao quát được tập dữ liệu.
- Hiện tượng **quá khớp** xảy ra tại mô hình rất hợp lý, rất khớp với tập huấn luyện nhưng khi đem ra dự đoán với dữ liệu mới thì lại không phù hợp. Nguyên nhân có thể do ta chưa đủ dữ liệu để đánh giá hoặc do mô hình của ta quá phức tạp. Mô hình bị quá phức tạp khi mà mô hình của ta sử dụng cả những nhiễu lớn trong tập dữ liệu để học, dẫn tới mất tính tổng quát của mô hình.
- Mô hình **vừa khớp** (**Good Fitting**) nằm giữa 2 mô hình chưa khớp và quá khớp cho ra kết quả hợp lý với cả tập dữ liệu huấn luyện và các giá trị mới, tức là nó mang được tính tổng quát như hình 1 ở giữa phía trên. Lý tưởng nhất là khớp được với nhiều dữ liệu mẫu và cả các dữ liệu mới. Tuy nhiên trên thực tế được mô hình như vậy rất hiếm.

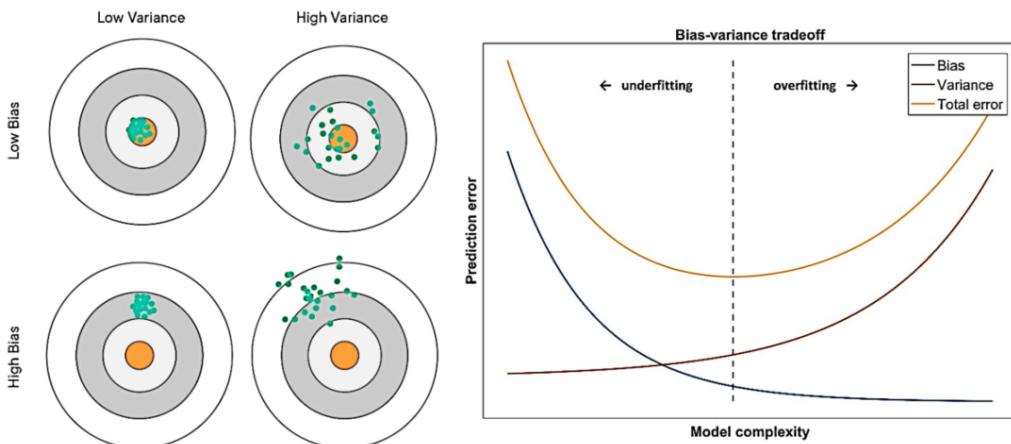
2.4.2 Độ lệch và phương sai

Trong học máy, sự cân bằng độ lệch (Bias) và phương sai (Variance) là thuộc tính của mô hình mà phương sai của các ước tính tham số giữa các

CHƯƠNG 2. MẠNG THẦN KINH NHÂN TẠO

mẫu có thể được giảm bớt bằng cách tăng độ chêch trong các tham số ước tính. Vấn đề độ lệch-phương sai hay vấn đề thiên vị-phương sai là xung đột trong việc cố gắng giảm thiểu đồng thời hai nguồn lỗi này ngăn các thuật toán học có giám sát tổng quát hóa ngoài tập huấn luyện của chúng:

- Độ lệch là một lỗi từ các giả định sai lầm của thuật toán học tập. Độ chêch cao có thể khiến thuật toán bỏ lỡ các mối quan hệ có liên quan giữa các tính năng và đầu ra mục tiêu (trang bị thiếu). Để giảm thiểu độ lệch, ta có thể tăng kích thước mô hình, tăng số nút mạng, thay đổi kiến trúc mô hình, thêm đặc trưng mới,...
- Phương sai là một sai số từ độ nhạy đối với các dao động nhỏ trong tập huấn luyện. Phương sai cao có thể do thuật toán mô hình hóa nhiễu ngẫu nhiên trong dữ liệu huấn luyện (trang bị quá mức). Để giảm thiểu phương sai, ta có thể thêm nhiều dữ liệu, sử dụng kỹ thuật dropout, lựa chọn các đặc trưng phù hợp.

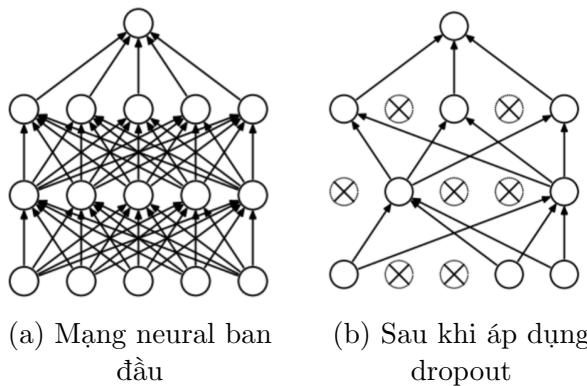


Hình 2.9: Sự đánh đổi giữa Bias và Variance

2.4.3 Kỹ thuật Dropout

Trong đề tài này sử dụng kỹ thuật dropout để tăng tính linh động và tổng quát hoá cho mô hình và chống quá khớp. Hiểu đơn giản kỹ thuật dropout là bỏ qua một số nút mạng theo tỷ lệ nào đó trong mỗi lần huấn luyện một epoch (bỏ qua không xem xét nó cả trong lan truyền xuôi lẫn trong lan truyền ngược chứ không phải bỏ ra khỏi mạng hoàn toàn).

Việc bỏ qua một số nút khiến cho các nút còn lại linh hoạt hơn cũng như mạnh mẽ hơn mà không phải phụ thuộc hoàn toàn vào các nút liên kết



Hình 2.10: Kỹ thuật dropout

với nó. Tuy nhiên, sử dụng kỹ thuật dropout sẽ làm tăng số epoch huấn luyện để đi đến hội tụ nhưng sẽ giảm thời gian huấn luyện. Trong kỹ thuật này, ta sử dụng hệ số dropout p đặc trưng cho tỷ số số lượng nút bị bỏ qua trên tổng số nút mạng. Thường thì người ta hay chọn hệ số dropout là $p = 0,5$.

Chương này đã trình bày khái quát hết kiến thức về mạng thần kinh nhân tạo, từ cấu trúc mạng đến phương pháp đánh giá hiệu xuất huấn luyện mạng thông qua việc sử dụng hàm lỗi và phương pháp tối thiểu hàm lỗi cũng như phương pháp cải thiện hiệu suất huấn luyện thông qua lan truyền ngược cũng như các vấn đề phát sinh trong quá trình huấn luyện như quá khớp, vừa khớp và chưa khớp.

Chương 3

MÔ HÌNH MẠNG NEURAL CHO CHATBOT

Ở chương này, ta đề cập đến xây dựng mạng đơn giản. Đồng thời tiến hành xây dựng tập huấn luyện đầu vào cho mạng để từ đó, ta đánh giá kết quả huấn luyện mạng và phương pháp tách lọc thông tin kết quả dự đoán của mạng để tìm câu trả lời thông qua các giải thuật và hệ cơ sở đã có.

3.1 Tổng quan về Chatbot

Thành phần học sâu: thành phần này có nhiệm vụ xử lý thông tin theo yêu cầu đưa vào và đưa ra câu trả lời thích hợp. Hai mô hình thường được dùng trong phần này là:

- **Retrieval Based:** sử dụng các bộ các câu trả lời đã được chuẩn bị trước và các thuật toán tìm kiếm, xử lý yêu cầu để chọn ra câu trả lời trong các câu trả lời có sẵn. Kỹ thuật tìm kiếm có thể ứng dụng máy học để phân loại yêu cầu, từ đó đưa ra câu trả lời phù hợp nhất.
- **Generative Model:** tự xử lý yêu cầu và tự sản sinh ra câu trả lời dựa trên các kỹ thuật máy học.

Trong đề tài này ta sử dụng mô hình Retrieval Based vì các thông tin, câu trả lời thường có tính cố định, lặp lại, đòi hỏi độ chính xác cao. Còn mô hình Generative Model chưa phù hợp vì sẽ làm phức tạp hoá hệ thống xây dựng, có thể gây mất thời gian xử lý hơn, đồng thời cũng có thể tạo ra các lỗi cú pháp cũng như lỗi nội dung, gây ảnh hưởng đến độ tin cậy của câu trả lời gửi đến người hỏi.

CHƯƠNG 3. MÔ HÌNH MẠNG NEURAL CHO CHATBOT

Với hệ thống xử lý trong chatbot ở đề tài này sẽ được trình bày đơn giản vai trò của các khối qua sơ đồ sau:



Hình 3.1: Hệ thống xử lý trong Chatbot

- **Khối NLP:** Tiến hành loại bỏ dấu câu, tách từ (tokenize), chuyển từ về dạng nguyên thuỷ (lemmatize), chuyển về dạng chữ thường (lower) và tiến hành xử lý câu văn vào thành dữ liệu thích hợp làm đầu vào cho khối ANN.
- **Khối ANN:** Thực hiện dự đoán câu văn đầu vào ở nhóm chủ đề và dự đoán chủ đề nào trong nhóm đó dựa trên xác suất.
- **Khối xử lý và tìm kiếm:** Xử lý và thu thập thông tin kết quả đầu ra của khối ANN, sử dụng các thuật toán và cơ sở dữ liệu để lấy ra câu trả lời thích hợp nhất.

3.2 Xây dựng và huấn luyện mạng neural

Trước khi đi vào quá trình xử lý trong chatbot, ta cần xác định quá trình xây dựng mạng neural trong khôi ANN. Quá trình gồm 3 bước: xây dựng tập huấn luyện; xây dựng cấu trúc và thông số cho mạng neural; huấn luyện, kiểm thử và lưu lại mô hình.

3.2.1 Xây dựng huấn luyện

Trước tiên, ta xây dựng các file json (JavaScript Object Notation) để lưu liệu theo cấu trúc sau:

CHƯƠNG 3. MÔ HÌNH MẠNG NEURAL CHO CHATBOT

```
{"tên nhóm chủ đề": [
    {"tag" : "tên chủ đề 1",
     "patterns" : ["từ khóa 1", "từ khóa 2", ... , "từ khóa n"],
     "response" : "câu trả lời (nếu cần thiết)"
    },
    {"tag" : "tên chủ đề 2",
     "patterns" : ["từ khóa 1", "từ khóa 2", ... , "từ khóa n"],
     "response" : "câu trả lời (nếu cần thiết)"
    },
    ...
    {"tag" : "tên chủ đề n",
     "patterns" : ["từ khóa 1", "từ khóa 2", ... , "từ khóa n"],
     "response" : "câu trả lời (nếu cần thiết)"
    }
]}]
```

Hình 3.2: Cấu trúc file json.

Sau khi có dữ liệu về nhóm chủ đề và các chủ đề cụ thể, ta triển khai quá trình tác từ bằng cách dùng hàm *word_tokenize* trong thư viện nltk của python. Công việc tách từ là xác định một câu văn bản rồi tách nó ra thành tập các từ hoặc dấu câu chứa trong đó. Chương trình sẽ lấy dữ liệu trong tất cả các "patterns" (trong file json) trong mỗi chủ đề, tách từ và lưu trữ tất cả kết quả vào một danh sách "words". Đồng thời danh sách tên "documents" sẽ lưu lại dữ liệu liên kết quả tách từ và kèm theo tên chủ đề ở mỗi từ khoá chủ đề tương ứng. Một danh sách "classes" sẽ lưu lại tất cả tên các chủ đề có trong nhóm chủ đề đang xét.

Tiếp theo ta lọc dấu câu, chuyển từ về dạng nguyên thể trong danh sách "words" bằng hàm *lemmatizer* trong thư viện nltk. Việc chuyển về dạng nguyên thể chỉ áp dụng với các ngôn ngữ có biến thể như tiếng Anh, tiếng Pháp, tiếng Đức,... còn đối với tiếng Việt thì không có sự biến đổi dạng từ, ví dụ như từ "**biết**", trong quá khứ là "**đã biết**", còn trong tiếng anh là **know**, trong quá khứ là **knew** hoặc have **known**. Vậy nên ta có thể bỏ qua quá trình này đối với dữ liệu là tiếng việt.

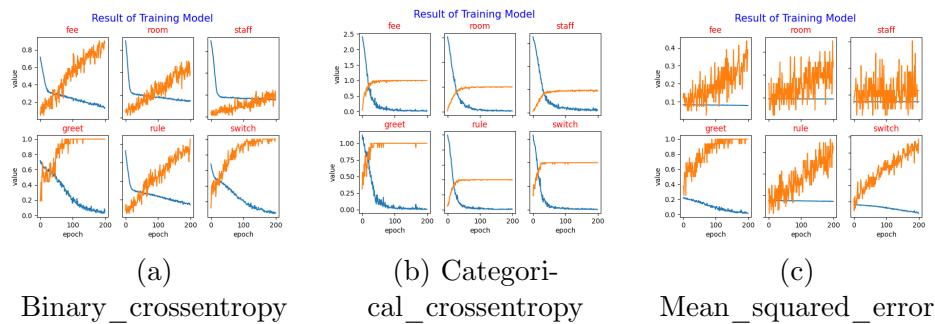
Sau đó, ta triển khai sắp xếp danh sách "words", sau đó tiến hành nhúng danh sách "words" và "document" vào file ".pkl" (Python Pickle File - là

một loại Developer Files) nhằm giữ nguyên trạng thái của đánh sách dưới dạng một luồng dài các byte. Điều này có tác dụng khi lấy dữ liệu ra sử dụng sẽ không bị biến đổi trạng thái, thích hợp cho xử lý chuỗi văn bản theo yêu cầu đưa tới khi tiến hành dự đoán bằng mạng neural

Ta tiến hành tạo tập huấn luyện cho mạng. Trước tiên ta tạo danh sách "train" rỗng chứa dữ liệu đầu vào và nhãn tương ứng với nó. Một danh sách "output_empty" có các giá trị đều bằng 0 và kích thước bằng với số lượng chủ đề trong nhóm chủ đề. Ta tạo danh sách "bag" rỗng và "pattern_words" chứa các từ khoá cụ thể trong mỗi đề tài, so với danh sách các từ trong "words" và nếu trùng thì lưu trong "bag". Tiếp theo ta tạo danh sách đầu ra "output_row" bằng với kích thước của "output_empty" và so sánh với chủ đề đang xét. Cuối cùng đưa dữ liệu "bag" và "output_row" vào "train".

3.2.2 Thiết kế xây dựng và huấn luyện mạng neural

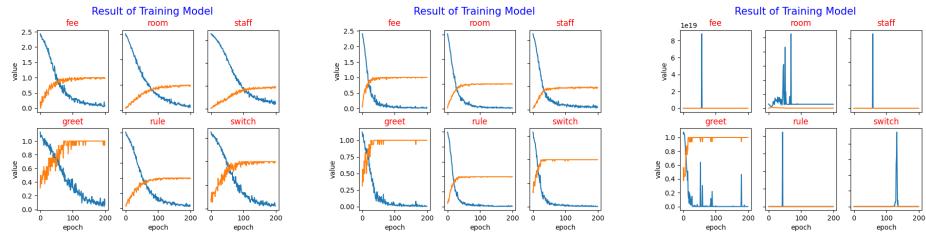
Công việc đầu tiên là lựa chọn hàm mất mát cho mạng neural. Huấn luyện với 3 hàm mất mát phổ thông ta cho kết quả như sau:



Hình 3.3: Lựa chọn hàm mất mát.

Dễ dàng thấy hàm mất mát Categorical_crossentropy phù hợp với mô hình này hơn cả. Tiếp theo, lựa chọn hệ số đà và hệ số học cho phương pháp tối ưu SGD:

CHƯƠNG 3. MÔ HÌNH MẠNG NEURAL CHO CHATBOT

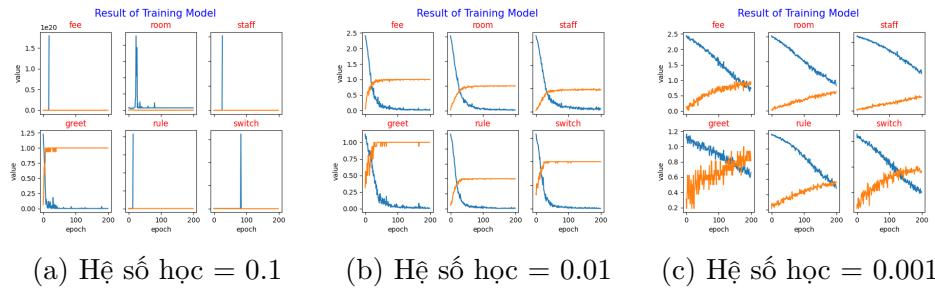


(a) Hệ số đà = 0.45

(b) Hệ số đà = 0.85

(c) Hệ số đà = 0.975

Hình 3.4: Lựa chọn hệ số đà.



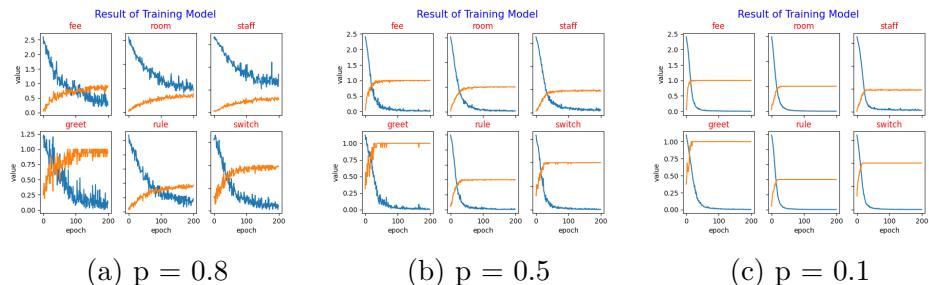
(a) Hệ số học = 0.1

(b) Hệ số học = 0.01

(c) Hệ số học = 0.001

Hình 3.5: Lựa chọn hệ số học.

Kỹ thuật dropout để tăng tính linh động và tổng quát hoá cho mô hình và chống quá khớp. Thông thường người ta hay chọn hệ số dropout $p = 0.5$. Tuy nhiên ta vẫn thử với vài thông số khác để xem kết quả mang lại như thế nào. Việc bỏ qua một số nút khiến cho các nút còn lại linh hoạt hơn cũng như mạnh mẽ hơn mà không phải phụ thuộc hoàn toàn vào các nút liên kết với nó.



(a) $p = 0.8$

(b) $p = 0.5$

(c) $p = 0.1$

Hình 3.6: Lựa chọn hệ số học.

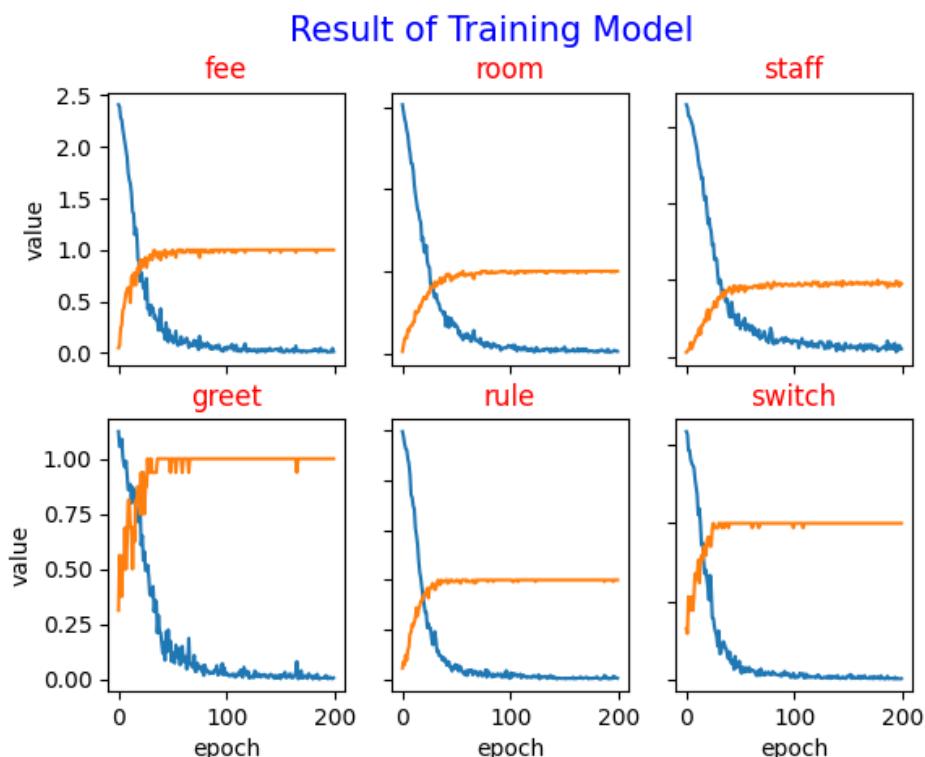
Từ các kết quả trên, quá trình thiết kế mạng neural nhân tạo như sau:

- Dữ liệu đầu vào gồm một số danh sách có số phần tử bằng kích thước của tập "words" có số chiều là 1. Lớp đầu vào có 256 nút mạng với

hàm kích hoạt là "ReLU". Lớp giữa gồm 128 lớp mạng với hàm kích hoạt là "ReLU". Lớp đầu ra có số nút mạng bằng với số chủ đề trong tập chủ đề (ở đây là 5 chủ đề) và hàm kích hoạt đầu ra là "softmax".

- Sử dụng hàm mất mát có dạng "categorical-crossentropy" với phương pháp tối ưu SGD + NAG với hệ số đà là 0.82, hệ số học là 0.01 và hệ số delay 10^{-8} . Huấn luyện 200 lần với hệ số batch-size =5. Đánh giá mạng dựa trên số dự đoán đúng trên độ chính xác (accuracy) và giá trị hàm loss.

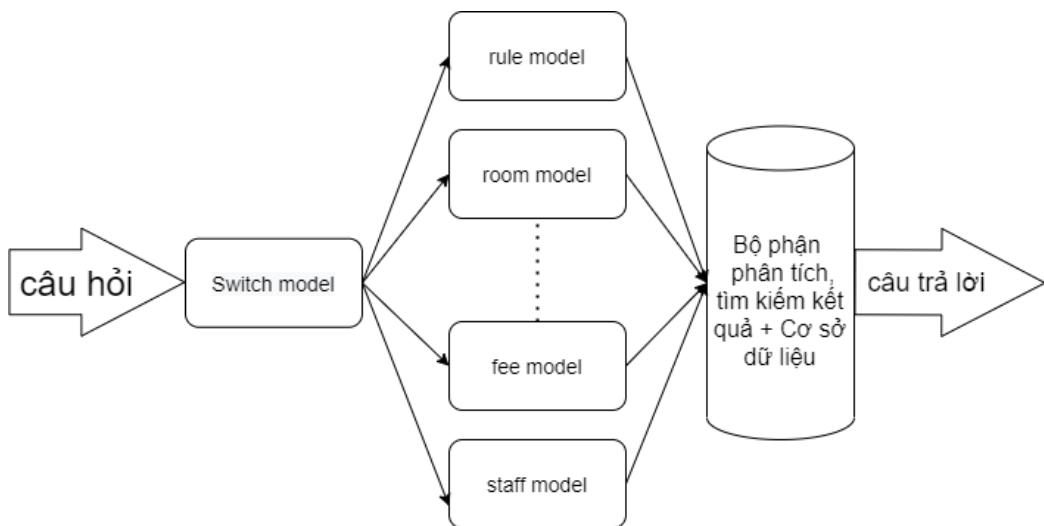
Tiến hành huấn luyện mạng neural, ta có kết quả huấn luyện như hình dưới đây. Dựa vào kết quả đó, ta nhận thấy các model cho độ chính xác cao gần bằng 1 và độ lỗi gần như bằng 0. Với kết quả như vậy, việc phân luồng và chọn chủ đề sẽ gần như chính xác với yêu cầu từ đầu vào. Về giải thuật phân luồng cũng như giải thuật tìm kiếm sẽ được trình bày ở phần sau.



Hình 3.7: Kết quả huấn luyện các model

3.3 Xử lý đầu vào và phân luồng theo yêu cầu

Sau khi xây dựng và huấn luyện các model, ta bắt tay vào xử lý dữ liệu đầu vào để khớp với đầu vào của mạng neural. Quá trình xử lý dữ liệu đầu vào cũng giống như quá trình tạo lập tập huấn luyện cho mạng neural. Trước hết, ta tách từ trong câu, lọc dấu câu và chuyển đổi thành chữ thường. Tiếp theo, ta tạo một chỗ để gom những từ vừa có trong câu và có trong danh sách "works" đã lưu ở dạng pickle. Sau đó chuyển những từ đó về dạng mạng nhị phân và làm đầu vào cho model switch có chức năng đưa các yêu cầu về đúng model chủ đề của nó. Ở đây, hệ thống gồm có các model:



Hình 3.8: Mô hình phân luồng câu hỏi và tìm kiếm câu trả lời

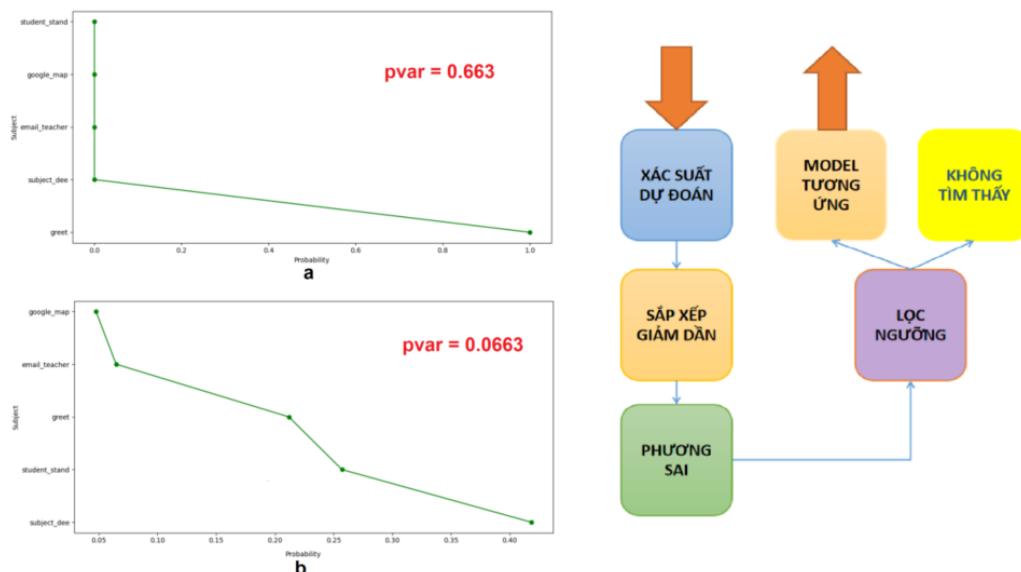
- **greet:** Mô hình này đơn giản là trả lời những câu đơn giản như chào hỏi, cảm ơn hay tạm biệt để tạo cảm giác thân thiện và tự nhiên hơn.
- **fee:** Mô hình này giúp ta trả lời những câu hỏi liên quan đến các loại phí dịch vụ có trong Ký túc xá.
- **room:** Mô hình này giúp ta trả lời những câu hỏi liên quan đến các loại phòng và vị trí của nó trong Ký túc xá.
- **staff:** Mô hình này giúp ta trả lời những câu hỏi liên quan đến các cán bộ đang công tác trong Ký túc xá.
- **rule:** Mô hình này giúp ta trả lời những câu hỏi liên quan đến các quy định trong Ký túc xá.

CHƯƠNG 3. MÔ HÌNH MẠNG NEURAL CHO CHATBOT

Quá trình phân luồng hoặc trả về kết quả theo từng model như sau: Trước tiên, dữ liệu được ta đưa vào model, sau đó sắp xếp các xác suất dự đoán chủ đề nhỏ trong các nhóm chủ đề theo thứ tự tăng dần. Sau đó ta tính toán phương sai theo chủ đề có xác suất lớn nhất. Dưới đây là công thức tính phương sai (độ lệch của dữ liệu so với giá trị lớn nhất của dự đoán của mô hình).

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - x_{max})^2 \quad (3.1)$$

Mô hình có phương sai càng lớn thì độ chính xác của xác suất dự đoán càng cao, do đó xác suất dự đoán chỉ tập trung ở một hoặc hai chủ đề xác định chứ không trải đều ở tất cả chủ đề. Nếu đầu vào model switch là "cảm ơn nha" thì ta dễ dàng biết thuộc chủ đề "greeting" nên phương sai *pvar* của mô hình tương đối lớn (hình a). Còn nếu đầu vào là "dịch hoài khổ quá" thì sẽ không thuộc bất cứ chủ đề nào trong "switch" nên phương sai *pvar* của mô hình tương đối nhỏ. Sau khi có kết quả phương sai, ta xác định ngưỡng để ta chấp nhận mô hình nào sẽ xử lý cho yêu cầu đầu vào. Trong đề tài này, ta chọn mức ngưỡng là 0.5.



Hình 3.9: Kết quả phương sai cho model switch

Ở các mô hình tiếp theo, ta cũng thực hiện tương tự như trên, tuy nhiên ta giảm ngưỡng phương sai còn 0.25 do ở model switch ta cần sự chính sách lớn trong quá trình lựa chọn để phân luồng, còn ở các model sau phân luồng, do trong tập train có những dữ liệu bị nhiễu như "phòng 4 người thường",

CHƯƠNG 3. MÔ HÌNH MẠNG NEURAL CHO CHATBOT

"phòng 4 người dịch vụ", "phòng 6 người thường",... dù đã cố gắng giảm nhiễu nhưng cũng không thể tránh khỏi sự sai sót. Nếu chọn ngưỡng 0.5 trở lên, các phương sai có thể có kết quả dưới 0.5, do đó ta tạm chấp nhận ngưỡng phương sai là 0.25.

Sau khi có kết quả dự đoán của mô hình con, ta tiến hành lựa chọn thông tin từ kết quả và trích xuất câu trả lời phù hợp. Câu trả lời ở đây được chuẩn bị theo hai dạng:

- **Câu hỏi có tính hệ thống:** cấu trúc câu trả lời sẽ được chuẩn bị sẵn kết hợp với dữ liệu lưu trong cơ sở dữ liệu (sẽ nói rõ hơn ở chương sau) để đưa ra câu trả lời hệ thống.

Ví dụ ta sử dụng cấu trúc câu trả lời hệ thống:

Giá dịch vụ: " + data1 + "là" + data2 + "/tháng"

sẽ cho ra hàng loạt câu trả lời: Giá dịch vụ: Gửi xe đạp là 30.000/tháng; Giá dịch vụ: Thuê tủ lạnh là 200.000/tháng;...

- **Câu hỏi có tính đặc thù, riêng biệt:** ta sẽ chuẩn bị một vài câu trả lời trong phần "reponse" trong file json rồi lựa chọn ngẫu nhiên 1 câu trả lời, để tránh gây sự buồn chán, tăng thêm "tính người" cho chatbot.

Chương này đã trình bày được cách thức xây dựng và huấn luyện mạng neural theo phương pháp Retrieval Based, sử dụng hàm mất mát có dạng "categorical-crossentropy" với phương pháp tối ưu SGD với hệ số đà là 0.85, hệ số học là 0.01. Huấn luyện 200 lần với hệ số batch-size = 5. Tiếp đó là phương pháp sử lý đầu vào dựa trên tỷ lệ xác suất đặc trưng đầu vào khớp với các đặc trưng trong mạng neural cũng như cách hệ thống trả về câu trả lời.

Chương 4

HỆ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG MÁY CHỦ WEB

Ở chương này, ta tìm hiểu về việc xây dựng hệ cơ sở dữ liệu thông qua MySQL và hệ thống máy chủ web dựa trên các máy chủ: máy chủ Caddy, máy chủ Gunicorn, máy chủ Flask,... để đưa ứng dụng web lên mạng Internet công cộng.

4.1 Cơ sở dữ liệu theo DBMS

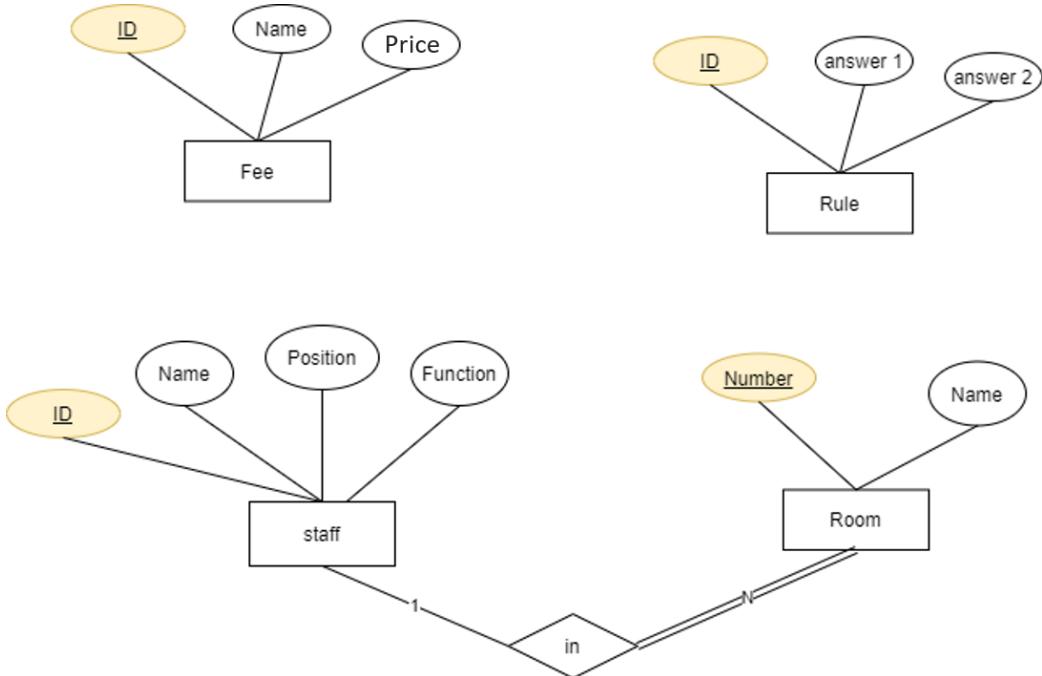
Đầu tiên, ta sẽ tìm hiểu tổng quan của cách tiếp cận cơ sở dữ liệu (CSDL) theo dạng hệ cơ sở dữ liệu (DBMS). Từ đó sẽ cho thấy được lợi ích khi ta sử dụng phương pháp này. Hiện nay, người ta thường dùng một số ngôn ngữ như: SQL (Structured Query Language), R,... để giao tiếp giữa DBMS và Database. Trong đề tài này, ta sử dụng phần mềm DBMS là MySQL để lưu trữ dữ liệu và ngôn ngữ là SQL. Ta lựa chọn phần mềm này vì dễ sử dụng, độ bảo mật cao, cộng đồng lớn, mã nguồn mở, đa năng, tốc độ xử lý cao, và được tích hợp Apache, PHP,... MySQL là một hệ thông quản trị cơ sở dữ liệu hoạt động theo mô hình khách chủ, quản lý dữ liệu thông qua các cơ sở dữ liệu. Mỗi CSDL có thể có nhiều bảng quan hệ chứa dữ liệu.

4.1.1 Lược đồ cơ sở dữ liệu

Hiện thực bài toán bằng mô hình quan hệ thực thể (ER) với **Yêu cầu thiết kế cơ sở dữ liệu cho hệ cơ sở dữ liệu:** Thiết kế một hệ cơ sở dữ liệu quản lý thông tin cán bộ, thông tin các phòng ban trong KTX cũng như các loại phí theo quy định của KTX.

Từ các yêu cầu trên, ta xây dựng sơ đồ ER như sau:

CHƯƠNG 4. HỆ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG MÁY CHỦ WEB



Hình 4.1: Sơ đồ quan hệ thực thể

4.1.2 Xây dựng cơ sở dữ liệu trong MySQL

Tổng quan về SQL MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (RDBMS) hoạt động theo mô hình client-server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu theo hình thức quản lý các mối liên hệ giữa chúng. Máy tính cài đặt và chạy phần mềm RDBMS được gọi là client (máy khách). Mỗi khi chúng cần truy cập dữ liệu, chúng kết nối tới máy chủ (server) RDBMS. Cách thức này chính là mô hình “client-server”.

MySQL Server là máy tính hay một hệ các máy tính cài đặt phần mềm MySQL dành cho server để giúp bạn lưu trữ dữ liệu trên đó, để máy khách có thể truy cập vào quản lý. Dữ liệu này được đặt trong các bảng, và các bảng có mối liên hệ với nhau. MySQL server nhanh, an toàn, đáng tin cậy. Phần mềm MySQL cũng miễn phí và được phát triển, phân phối và hỗ trợ bởi Oracle Corporation.

MySQL client không cần phải cài phần mềm MySQL của Oracle mà là nói chung của mọi phần mềm có thể thực hiện truy vấn lên một MySQL server và nhận kết quả trả về. MySQL client điển hình là đoạn mã PHP script trên một máy tính hay trên cùng server dùng để kết nối tới cơ sở dữ liệu MySQL database. Phpmyadmin cũng là một MySQL client có giao diện người dùng.

CHƯƠNG 4. HỆ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG MÁY CHỦ WEB

Một số công cụ miễn phí dùng làm MySQL là: MySQL Workbench (Mac, Windows, Linux), Miễn phí, mã nguồn mở; Sequel Pro (Mac), miễn phí, mã nguồn mở; HeidiSQL (Windows; chạy trên Mac hoặc Linux bằng WINE emulator), miễn phí; phpMyAdmin (web app), miễn phí, mã nguồn mở;...

Trong môi trường MySQL, máy khách (client) và máy chủ (server) tương tác qua lại với nhau theo nguyên lý:

- MySQL tạo ra các bảng nhằm lưu trữ dữ liệu, đồng thời định nghĩa mối quan hệ giữa các bảng đó.
- Client gửi các yêu cầu SQL bằng lệnh đặc biệt lên MySQL.
- Ứng dụng trên server nhận được và phản hồi thông tin, trả kết quả về máy khách.

MySQL không phải là hệ quản lý cơ sở dữ liệu (DBMS) duy nhất trên thị trường, nhưng nó đích thực phổ biến nhất và chỉ xếp sau Oracle Database khi xét đến những thông số chính như số lượng tìm kiếm, profile người dùng trên LinkedIn, và lượng thảo luận trên các diễn đàn Internet. Lý do chính vì sao rất nhiều ông lớn công nghệ phụ thuộc vào MySQL là gì? Các lý do quan trọng như sau:

- **Linh hoạt và dễ dùng:** MySQL đơn giản, dễ sử dụng. Ngoài ra, phần mềm này có thể hoạt động trên khá nhiều hệ điều hành nhằm cung cấp nhiều hàm tiện ích mạnh mẽ.
- **Hiệu năng cao:** Nhiều server clusters sử dụng MySQL. Bất kể lưu trữ dữ liệu lớn của các trang thương mại điện tử hoặc những hoạt động kinh doanh nặng nề liên quan đến công nghệ thông tin, MySQL cũng có thể đáp ứng được với tốc độ cao, mượt mà.
- **Tiêu chuẩn trong ngành:** Ngành công nghệ và dữ liệu đã sử dụng MySQL nhiều năm, vì vậy nó là một kỹ năng căn bản một chuyên gia lập trình. Người dùng MySQL cũng có thể triển khai dự án nhanh và thuê các chuyên gia dữ liệu với mức phí nếu họ cần.
- **Bảo mật cao:** An toàn dữ liệu luôn là vấn đề quan trọng nhất khi chọn phần mềm RDBMS. Với hệ thống phân quyền truy cập và quản lý tài khoản, MySQL đặt tiêu chuẩn bảo mật rất cao. Mã hóa thông tin đăng nhập và chứng thực từ host đều khả dụng.
- **Có thể khôi phục dữ liệu:** MySQL cho phép người dùng khôi phục dữ liệu, tránh khỏi ảnh hưởng của các sự cố.

CHƯƠNG 4. HỆ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG MÁY CHỦ WEB

Tuy nhiên, các nhược điểm của MySQL cũng nên được cân nhắc trước khi quyết định liệu có nên dùng nó hay không.

- **Bị giới hạn:** MySQL bị hạn chế về một vài tính năng mà các ứng dụng có thể sẽ cần đến
- **Độ tin cậy không quá cao:** So với các hệ quản trị CSDL quan hệ khác, độ tin cậy của MySQL không quá cao.
- **Bị hạn chế về dung lượng:** Số bản ghi trong MySQL càng tăng thì truy xuất dữ liệu càng trở nên khó khăn do hạn chế về dung lượng.

Trong luận văn này, MySQL đã được thiết lập tại cổng 3306 trên máy chủ web. Cơ sở dữ liệu trong MySQL đã kết nối với ứng dụng bằng thiết lập:

```
def dbconfig(app):
    mysql = MySQL()
    app.config['MYSQL_DATABASE_USER'] = 'root'
    app.config['MYSQL_DATABASE_PASSWORD'] = '*****'
    app.config['MYSQL_DATABASE_DB'] = 'cb'
    app.config['MYSQL_DATABASE_HOST'] = '0.0.0.0'
    app.config['MYSQL_DATABASE_PORT'] = 3306
    mysql.init_app(app)
    return mysql
```

Hình 4.2: Cú pháp kết nối ứng dụng với MySQL

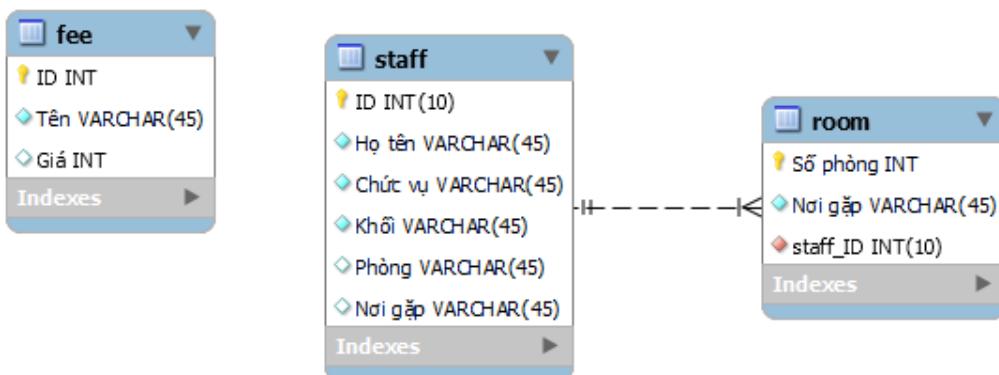
Nhà khoa học máy tính Ted Codd tạo ra SQL vào đầu thập niên 1970s với IBM theo mô hình liên hệ. Nó được sử dụng rộng khắp vào năm 1974 và nhanh chóng thay thế các ngôn ngữ lõi thời khác như ISAM và VISAM. SQL nói server phải làm gì với dữ liệu. Lệnh SQL sẽ nói cho server biết cần phải thực hiện một số hành động:

1. Truy vấn dữ liệu – data query: yêu cầu một thông tin cụ thể trong database có sẵn.
2. Điều hành dữ liệu – data manipulation: thêm, xóa, sửa, sắp xếp, và các tác vụ điều hành khác để chỉnh sửa dữ liệu, giá trị, và cách hiển thị.
3. Loại dữ liệu – data identity: định nghĩa loại dữ liệu, như là đổi từ dữ liệu số thành số nguyên. Nó cũng bao gồm định nghĩa một schema hoặc định nghĩa mối quan hệ của từng bảng trong cơ sở dữ liệu

4. Quyền truy cập data – data access control: cung cấp biện pháp bảo mật để bảo vệ dữ liệu, bao gồm việc quyết định ai có thể xem thông tin dữ liệu nào trong database

4.1.3 Kết quả xây dựng bảng

Trước khi xây dựng các bảng trong CSDL, ta phải định nghĩa kiểu dữ liệu và các ràng buộc cho các trường trong các bảng. Các ràng buộc này góp phần đảm bảo tính cấu trúc, tính toàn vẹn và tính nhất quán trong CSDL. Để thiết lập ràng buộc, ta dùng lệnh CONSTRAINTS trong MySQL. Các ràng buộc cơ bản như: PRIMARY KEY (khoá chính), FOREIGN KEY (khoá phụ), NOT NULL (không trống), CHECK (giới hạn phạm vi dữ liệu), UNIQUE (độc nhất), DEFAULT (dữ liệu mặc định của mỗi trường khi tạo ra) cho mỗi trường. Sau đó tiến hành nhập dữ liệu cho bảng. Ta cần đảm bảo tính đúng đắn của kiểu dữ liệu cũng như những ràng buộc của các trường trong bảng, nếu không ta sẽ không thể thêm hàng dữ liệu đó vào bảng. Từ các bảng và các ràng buộc, ta có lược đồ CSDL trong MySQL như sau:



Hình 4.3: Lược đồ cơ sở dữ liệu được xây dựng từ MySQL

4.2 Máy chủ web

4.2.1 Tổng quan

Để thực hiện luận văn này, một máy chủ đảm máy hệ điều hành Ubuntu với RAM là 1Gb đã được sử dụng từ website: <https://my.cloudfly.vn/>. Cấu hình chi tiết máy chủ như sau:

CHƯƠNG 4. HỆ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG MÁY CHỦ WEB

```
PS C:\Users\Ly Kien> ssh root@103.82.27.5
root@103.82.27.5's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-77-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:      https://landscape.canonical.com
 * Support:         https://ubuntu.com/advantage

 System information as of Fri 02 Jul 2021 03:33:51 PM +07

 System load:  0.0          Processes:           116
 Usage of /:   39.7% of 19.99GB  Users logged in:     0
 Memory usage: 44%          IPv4 address for eth0: 103.82.27.5
 Swap usage:   15%

 * Super-optimized for small spaces - read how we shrank the memory
 footprint of MicroK8s to make it the smallest full K8s around.

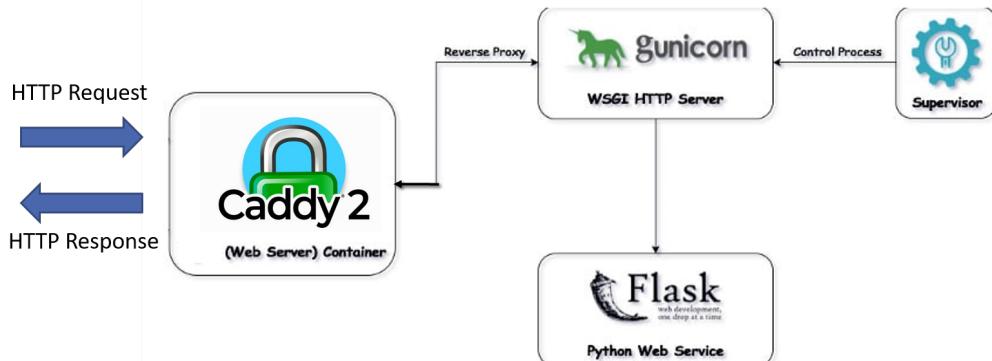
 https://ubuntu.com/blog/microk8s-memory-optimisation
```

Hình 4.4: Cấu hình máy chủ server.

Hệ thống máy chủ server gồm 4 thành phần chính:

- **Flask**: chương trình ứng dụng (application program).
- **Gunicorn**: máy chủ ứng dụng (application server).
- **Supervisor**: quản lý quy trình.
- **Caddy**: máy chủ web giao diện công khai (public-facing webserver).

Ta sẽ tìm hiểu kỹ hơn về các thành phần này ở các phần tiếp theo.



Hình 4.5: Lực đồ liên hệ giữa các thành phần trong hệ thống máy chủ Web.

4.2.2 Flask Server

Flask là một micro web framework được viết bằng Python. Nó được phân loại là một microframework vì nó không yêu cầu các công cụ hoặc thư viện cụ thể. Nó không có lớp trừu tượng cơ sở dữ liệu, xác thực biểu mẫu hoặc bất kỳ thành phần nào khác mà các thư viện bên thứ ba đã có từ trước cung cấp các chức năng phổ biến. Tuy nhiên, Flask hỗ trợ các tiện ích mở rộng có thể thêm các tính năng ứng dụng như thể chúng được triển khai trong chính Flask. Các tiện ích mở rộng tồn tại cho người lập bản đồ quan hệ đối tượng, xác thực biểu mẫu, xử lý tải lên, các công nghệ xác thực mở khác nhau và một số công cụ liên quan đến khuôn khổ chung.

Các thư viện python *flask* được sử dụng trong luận văn này như sau:

```
from flask import Flask, request, redirect, render_template
```

- Flask: tên gói được sử dụng để phân giải tài nguyên từ bên trong gói hoặc thư mục mà model được chứa trong đó tuỳ thuộc vào việc tham số gói phân giải thành gói python thực (thư mục tệp `__init__.py` bên trong) hay model chuẩn (chỉ là một tệp `.py`)
- request: để truy cập dữ liệu yêu cầu đến. Flask phân tích cú pháp dữ liệu yêu cầu đến cho bạn và cấp cho bạn quyền truy cập vào đó thông qua đối tượng `request` đó
- Redirect: trả về một đối tượng phản hồi, nếu được gọi, sẽ chuyển hướng client đến vị trí đích
- render_template: kết xuất template từ thư mục template

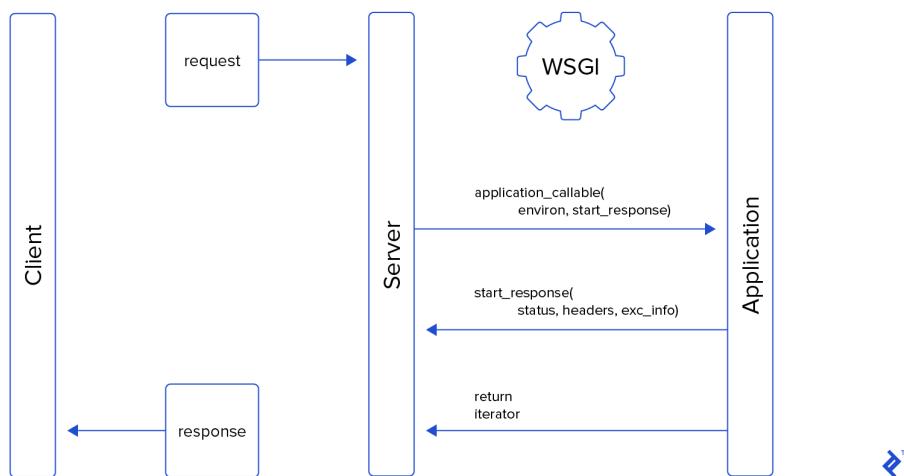
Để khởi tạo Flask Server, ta sử dụng lệnh `app = Flask(__name__)`, và thiết lập chạy server application trên webserver, ta sử dụng `app.run(host='0.0.0.0', port=8080, debug=True)`.

4.2.3 Gunicorn

Giao diện cổng máy chủ web (WSGI) là tiêu chuẩn để kết nối các ứng dụng web Python với các máy chủ web, đáp ứng các yêu cầu HTTP trong mã Python. Mỗi khi nhận được yêu cầu từ các máy khách HTTP hướng đến ứng dụng, một ứng dụng có thể gọi được ứng dụng sẽ được gọi bởi máy chủ / cổng. Cuối cùng, WSGI trở thành cách tiếp cận được chấp nhận để chạy các ứng dụng Python. Nó là máy chủ kích hoạt ứng dụng web và truyền thông tin liên quan, với chức năng gọi lại cho ứng dụng. Việc xử lý yêu cầu diễn ra ở phía ứng dụng, trong khi máy chủ nhận được phản hồi, bằng cách sử

CHƯƠNG 4. HỆ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG MÁY CHỦ WEB

dụng chức năng gọi lại. Các khung Python hỗ trợ WSGI là Django, web2py, **Flask**, TurboGears và CherryPy.



Hình 4.6: Cấu trúc hoạt động của WSGI

Gunicorn "Green Unicorn" là một Python Web Server Gateway Interface (WSGI) HTTP server . Nó là một mô hình pre-fork worker. Máy chủ Gunicorn tương thích rộng rãi với một số khuôn khổ web , được triển khai đơn giản, tốn ít tài nguyên máy chủ và khá nhanh. Gunicorn rất dễ sử dụng và hỗ trợ tốt cho nhiều web framework khác nhau như Django, Flask ... Các đặc điểm của Gunicorn:

- Có thể chạy bất kỳ ứng dụng và framework Python web nào theo chuẩn WSGI.
- Dễ dàng thay thế các web server trong môi trường phát triển sản phẩm (môi trường dành cho lập trình viên để kiểm tra hoạt động của các ứng dụng) mà không cần thay đổi mã nguồn ứng dụng.
- Hỗ trợ rất nhiều loại worker và các cấu hình cho từng loại worker.
- Hỗ trợ các loại worker đồng bộ và bất đồng bộ.
- Hỗ trợ SSL.
- Hỗ trợ python 2 và python 3.

Gunicorn được thực hiện theo mô hình UNIX pre-fork server:

CHƯƠNG 4. HỆ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG MÁY CHỦ WEB

- Khi khởi chạy, Gunicorn mở một tiến trình gốc (master process), tiến trình gốc này có thể được nhân bản (fork) thành các tiến trình con, các tiến trình con này được gọi là các worker.
- Vai trò của tiến trình gốc là đảm bảo số lượng các worker luôn luôn giống với con số đã được định nghĩa trong các file cấu hình hay trong tham số dòng lệnh. Nếu một worker vì vấn đề gì đó mà bị chết, tiến trình gốc sẽ tạo ra một worker mới bằng cách fork chính nó một lần nữa.
- Vai trò của các worker là tiếp nhận và xử lý các yêu cầu HTTP.
- Từ pre trong mô hình pre-fork nghĩa là tiến trình gốc tạo ra các worker trước khi xử lý bất kỳ yêu cầu HTTP nào.
- Kernel của hệ điều hành đảm nhận vai trò cân bằng tải giữa các worker

Worker - Tiến trình UNIX (workers)

Mỗi worker là một tiến trình UNIX, nó là một thực thể của ứng dụng, các worker là các thực thể riêng biệt của ứng dụng và chúng không chia sẻ tài nguyên bộ nhớ với nhau.

Số lượng worker hợp lý khi chạy Gunicorn trên một máy vật lý (máy ảo) thường là $(2 * number_{of_cpu}) + 1$.

Cho một máy vật lý có 4 CPU, 9 là số lượng worker hợp lý để chạy trên máy đó.

Các luồng (threads)

Ngoài việc cho phép tạo ra nhiều worker, gunicorn còn cho phép một worker có thể tạo ra nhiều thread, các thread trong một worker chia sẻ cùng tài nguyên bộ nhớ với nhau.

Để sử dụng thread với gunicorn, chúng ta sử dụng tham số threads. Số lượng tác vụ cùng lúc tối đa là $workers * threads = 36$ trong trường hợp ví dụ của chúng ta.

Tuy nhiên số lượng tác vụ được gợi ý nên dùng cho trường hợp dùng lẫn cả workers và threads vẫn là $(2 * number_{of_cpu}) + 1 = 9$ (trong trường hợp máy có 4 CPU). Do vậy để có được tối đa 9 tác vụ chạy đồng thời, ta có thể sử dụng 3 worker và mỗi worker có 3 thread.

Loại worker

Gunicorn hỗ trợ nhiều loại worker

- **Sync worker:** Loại worker này thường được sử dụng với ứng dụng không có thời gian đọc ghi vào ổ đĩa dài, hoặc không có những yêu cầu với thời gian lớn (long request - loại yêu cầu như này sẽ làm cho các yêu cầu khác phải đợi cho đến khi yêu cầu hiện tại được hoàn thành, điều này có thể làm cho các yêu cầu thất bại do kết nối bị lỗi timeout).
- **Asynchronous worker** bao gồm hai loại: gevent và eventlet, hai loại này được phát triển dựa trên thư viện Greenlet. Thư viện này cung cấp các phương thức giúp các tác vụ được xử lý đồng thời, nó giải quyết vấn đề của sync worker, cái mà chỉ xử lý được một yêu cầu tại một thời điểm.
- **Tornado worker:** Loại worker này được thiết kế để làm việc với python framework Tornado. Tornado framework là một thư viện mạng cung cấp giải pháp đọc ghi bất đồng bộ non-blocking. Nó xử lý tốt các yêu cầu với thời gian lớn (long request).
- **AsyncIO worker:** bao gồm hai loại gthread và gaiohttp.

gaiohttp sử dụng thư viện aiohttp. Thư viện này thực hiện việc nhập xuất bất đồng bộ trong môi trường mạng ở cả phía client lẫn server và đặc biệt nó hỗ trợ web socket rất tốt.

gthread giữ kết nối trong một nhóm thread và đợi cho đến khi có sự kiện, sự kiện đó sẽ được xử lý bởi một trong các thread ở trong nhóm.

Mỗi loại worker cung cấp các tính năng khác nhau để xử lý các yêu cầu. Cấu hình với lựa chọn (option) worker-class được sử dụng để chọn loại worker mà gunicorn. Giá trị của worker-class có thể là một trong số sau: sync, gevent, eventlet, tornado, gaiohttp, và gthread. Nếu không sử dụng lựa chọn worker-class thì giá trị mặc định của nó là sync.

4.2.4 Supervisor

Supervisor là một hệ thống máy khách/ máy chủ cho phép người dùng kiểm soát một số quy trình trên hệ điều hành giống UNIX. Supervisor được xem là một công cụ hữu ích để theo dõi và kiểm soát Gunicorn.

Các đặc trưng của Supervisor:

- **Dơn giản:** được định cấu hình thông qua tệp cấu hình kiểu INI đơn giản dễ học. Nó cung cấp nhiều tùy chọn cho mỗi quy trình giúp cuộc sống của bạn dễ dàng hơn như khởi động lại các quy trình bị lỗi và xoay vòng nhật ký tự động.

CHƯƠNG 4. HỆ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG MÁY CHỦ WEB

- Tập trung: cung cấp cho bạn một nơi để bắt đầu, dừng và giám sát các quy trình của bạn. Các quá trình có thể được kiểm soát riêng lẻ hoặc theo nhóm. Bạn có thể định cấu hình Supervisor để cung cấp giao diện web và dòng lệnh cục bộ hoặc từ xa.
- Có hiệu quả: Supervisor bắt đầu các quy trình con của nó thông qua fork/ execute và các quy trình con không daemonize. Hệ điều hành báo hiệu cho Supervisor ngay lập tức khi một quy trình kết thúc, không giống như một số giải pháp dựa vào các tệp PID rắc rối và thăm dò định kỳ để khởi động lại các quy trình không thành công.
- Có thể mở rộng: Supervisor có một giao thức thông báo sự kiện đơn giản mà các chương trình được viết bằng bất kỳ ngôn ngữ nào có thể sử dụng để giám sát nó và một giao diện XML-RPC để điều khiển. Nó cũng được xây dựng với các điểm mở rộng có thể được các nhà phát triển Python tận dụng.
- Tương thích: Supervisor hoạt động trên mọi thứ ngoại trừ Windows. Nó được thử nghiệm và hỗ trợ trên Linux, Mac OS X, Solaris và FreeBSD. Nó được viết hoàn toàn bằng Python (Python 3 phiên bản 3.4 trở lên và trên Python 2 phiên bản 2.7), vì vậy cài đặt không yêu cầu trình biên dịch C.

Các thành phần chính của Supervisor:

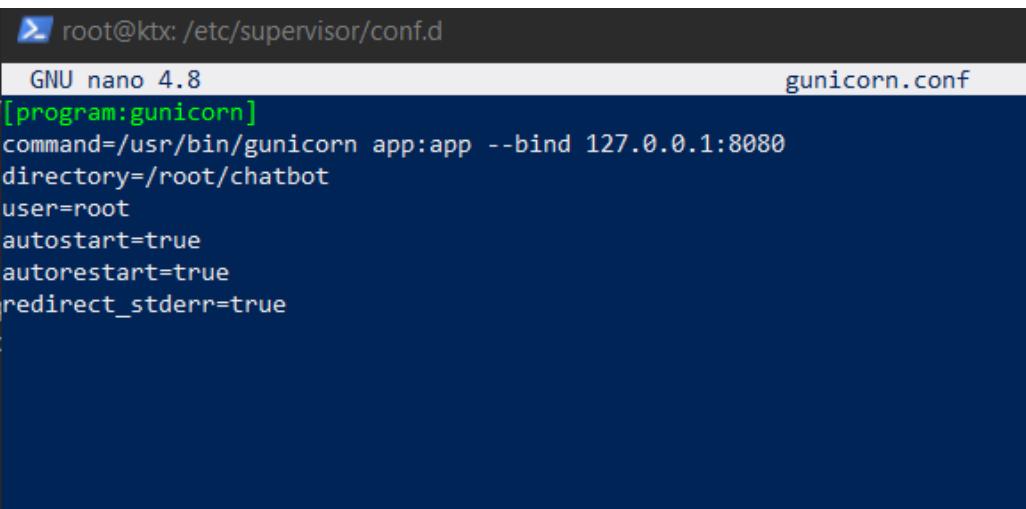
- **supervisord:** Phần máy chủ của Supervisor được đặt tên là supervisord. Nó chịu trách nhiệm khởi động các chương trình con theo lệnh gọi của chính nó, phản hồi các lệnh từ máy khách, khởi động lại các quy trình con bị lỗi hoặc đã thoát, ghi lại đầu ra stdout và stderr của quy trình con, đồng thời tạo và xử lý “sự kiện” tương ứng với các điểm trong vòng đời của quy trình con. Quá trình máy chủ sử dụng tệp cấu hình. Điều này thường nằm trong `/etc/supervisord.conf`. Tệp cấu hình này là tệp cấu hình kiểu “Windows-INI”. Điều quan trọng là phải giữ an toàn cho tệp này thông qua các quyền thích hợp của hệ thống tệp vì nó có thể chứa tên người dùng và mật khẩu không được mã hóa.
- **supervisorctl:** Phần dòng lệnh khách của trình giám sát được đặt tên là supervisorctl . Nó cung cấp một giao diện giống như shell với các tính năng được cung cấp bởi supervisord. Từ supervisorctl, người dùng có thể kết nối với các quy trình giám sát khác nhau (mỗi lần một), nhận trạng thái trên các quy trình con được kiểm soát bởi, dừng và khởi động các quy trình con của, và nhận danh sách các quy trình đang chạy của một giám sát. Máy khách dòng lệnh nói chuyện với máy chủ

CHƯƠNG 4. HỆ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG MÁY CHỦ WEB

qua ổ cắm miền UNIX hoặc ổ cắm internet (TCP). Máy chủ có thể khẳng định rằng người dùng của máy khách phải xuất trình thông tin xác thực trước khi nó cho phép anh ta thực hiện các lệnh. Quy trình máy khách thường sử dụng tệp cấu hình giống như máy chủ nhưng bất kỳ tệp cấu hình nào có phần [supervisorctl] trong đó sẽ hoạt động.

- **Máy chủ web:** Một giao diện người dùng web có chức năng tương đương với supervisorctl có thể được truy cập qua trình duyệt nếu bạn khởi động supervisord trên ổ cắm internet. Truy cập URL máy chủ để xem và kiểm soát trạng thái quy trình thông qua giao diện web sau khi kích hoạt phần [inet_http_server] của tệp cấu hình.
- **Giao diện XMLRPC:** Cùng một máy chủ HTTP phục vụ giao diện người dùng web cung cấp giao diện XML-RPC có thể được sử dụng để thám vấn và kiểm soát người giám sát cũng như các chương trình mà nó chạy.

Trong luận văn này, Gunicorn đã được cấu hình đơn giản với Supervisor ở cổng 8080.



```
root@ktx: /etc/supervisor/conf.d
GNU nano 4.8                                     gunicorn.conf
[program:gunicorn]
command=/usr/bin/gunicorn app:app --bind 127.0.0.1:8080
directory=/root/chatbot
user=root
autostart=true
autorestart=true
redirect_stderr=true
```

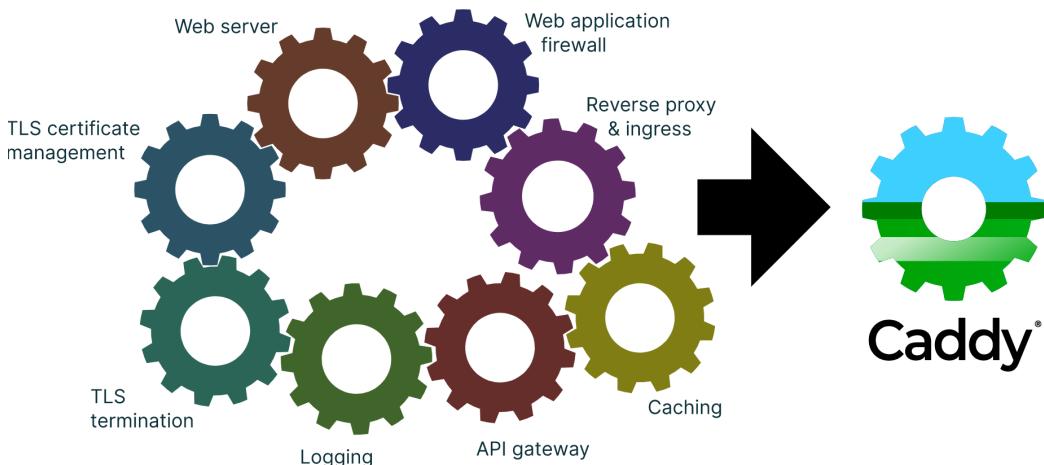
Hình 4.7: Cấu hình các quá trình của Gunicorn bằng Supervisor

4.2.5 Caddy

Caddy là một máy chủ của các máy chủ (hầu hết mọi người sử dụng Caddy như một máy chủ web hoặc proxy), được viết bằng ngôn ngữ Go. Với các modun cần thiết, nó có thể đảm nhận vai trò của bất kỳ quá trình xử lý dài nào. Caddy kích hoạt HTTPS theo mặc định cho các trang web có tên

CHƯƠNG 4. HỆ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG MÁY CHỦ WEB

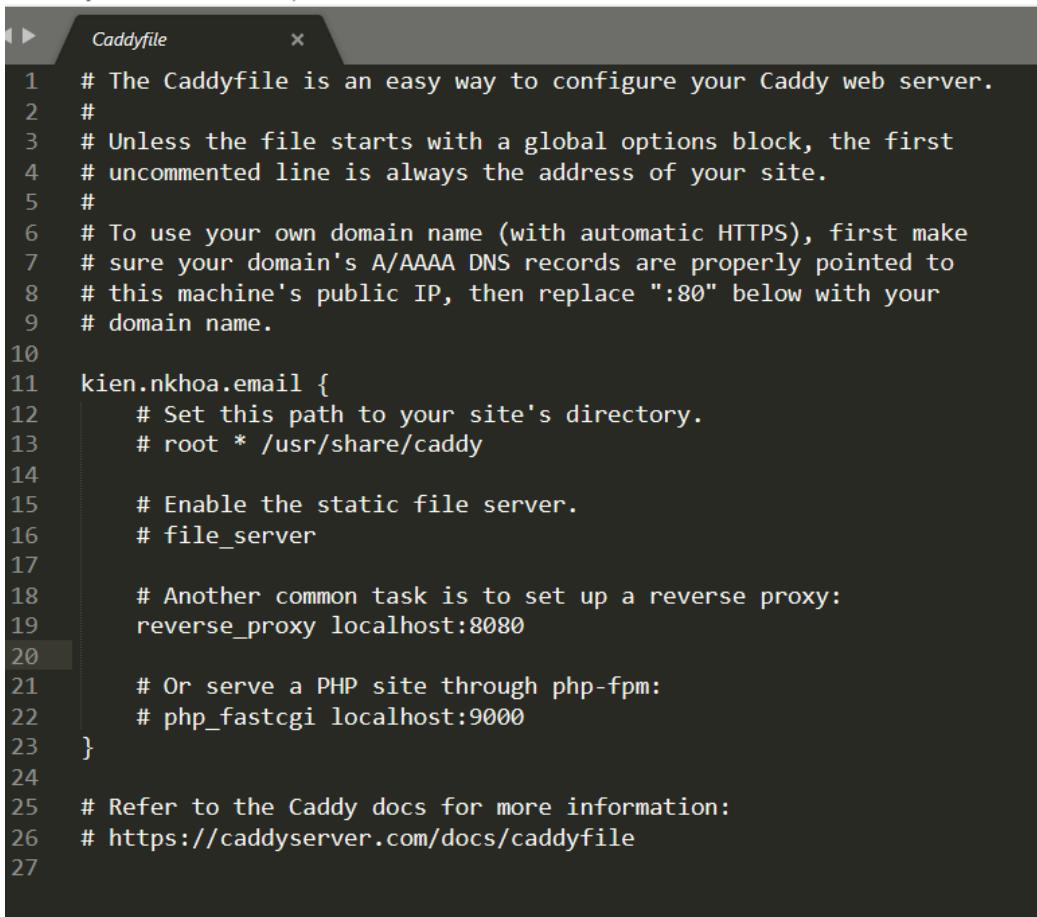
miền đủ điều kiện (tên mà chứng chỉ TLS có thể được thương lượng thông qua giao thức ACME) và chuyển hướng các yêu cầu HTTP đến HTTPS. [20] Nó nhận được các chứng chỉ khi cần thiết trong quá trình khởi động và giữ chúng được gia hạn trong suốt thời gian tồn tại của máy chủ. Let's Encrypt là cơ quan cấp chứng chỉ mặc định, nhưng người dùng có thể tùy chỉnh ACME CA được sử dụng, điều này thường cần thiết khi kiểm tra cấu hình. Tính đến quý 1 năm 2016, Caddie chiếm khoảng 2% số chứng chỉ do Let's Encrypt cấp. Một cấu hình thay thế cho phép Caddy chỉ nhận được chứng chỉ khi cần trong quá trình bắt tay TLS chứ không phải khi khởi động, một tính năng được gọi là "TLS theo yêu cầu". Để kích hoạt tính năng này, người dùng phải chỉ định số lượng chứng chỉ tối đa có thể được cấp theo cách này. Khi Caddy nhận được yêu cầu về một tên máy chủ mà nó chưa có chứng chỉ, nó sẽ thương lượng chứng chỉ mới qua ACME và phân phát nó ngay lập tức, đồng thời lưu chứng chỉ thu được vào bộ nhớ và lưu trữ trên đĩa. Quá trình này thường mất vài giây và phải tuân theo các giới hạn tỷ lệ chật chẽ. Khi có TLS, Caddie sẽ tự động xoay các khóa phiếu phiên theo định kỳ để giúp bảo mật cho các chuyển tiếp bí mật.



Hình 4.8: Caddy

Cấu hình vừa động vừa có thể xuất với API của Caddy. Mặc dù không yêu cầu tệp cấu hình, ta vẫn có thể sử dụng chúng; Định dạng của tài liệu cấu hình có nhiều dạng với bộ điều hợp cấu hình, nhưng ngôn ngữ cấu hình gốc của Caddy là JSON. Cách cấu hình Caddy yêu thích của hầu hết mọi người là sử dụng Caddyfile. Và trong luận văn này, Caddyfile cũng được sử dụng để cấu hình cho máy chủ.

CHƯƠNG 4. HỆ CƠ SỞ DỮ LIỆU VÀ HỆ THỐNG MÁY CHỦ WEB



```
Caddyfile
1 # The Caddyfile is an easy way to configure your Caddy web server.
2 #
3 # Unless the file starts with a global options block, the first
4 # uncommented line is always the address of your site.
5 #
6 # To use your own domain name (with automatic HTTPS), first make
7 # sure your domain's A/AAAA DNS records are properly pointed to
8 # this machine's public IP, then replace ":80" below with your
9 # domain name.
10
11 kien.nkhoa.email {
12     # Set this path to your site's directory.
13     # root * /usr/share/caddy
14
15     # Enable the static file server.
16     # file_server
17
18     # Another common task is to set up a reverse proxy:
19     reverse_proxy localhost:8080
20
21     # Or serve a PHP site through php-fpm:
22     # php_fastcgi localhost:9000
23 }
24
25 # Refer to the Caddy docs for more information:
26 # https://caddyserver.com/docs/caddyfile
27
```

Hình 4.9: Cấu hình máy chủ Web bằng Caddyfile.

Trong chương này đã trình bày cách thức xây dựng, quả lý hệ cơ sở dữ liệu với công cụ MySQL và quá trình 4 thành phần chính trên máy chủ web liên kết và hoạt động như thế nào. Ở chương sau sẽ trình bày cách thức để Chatbot giao tiếp người dùng.

Chương 5

PHƯƠNG PHÁP GIAO TIẾP NGƯỜI DÙNG

Ở chương này, ta sẽ tìm hiểu về cách thức truyền nhận thông tin giữa server và các client khác nhau. Ngoài ra chương này còn đề cập đến phương pháp giao tiếp với người dùng dưới dạng Facebook messenger và trong hộp chat trên trang web.

5.1 Giao thức HTTPS

5.1.1 Một số khái niệm

Ta cần HTTPS khi thiết lập botchat theo yêu cầu của Facebook messenger, nên hãy tìm hiểu qua một số khái niệm của HTTP và HTTPS.

HTTP (HyperText Transfer Protocol) là giao thức tầng ứng dụng của web, được chia sẻ trên cả hai phía chủ và khách. Các tiến trình chủ khách trên các hệ thống đầu/cuối khác nhau giao tiếp với nhau thông qua trao đổi thông điệp HTTP. HTTP quy định cấu trúc thông điệp cũng như cách trao đổi thông điệp giữa chủ với khách.

HTTPS (Hypertext Transfer Protocol Secure) là một phiên bản bảo mật của giao thức HTTP sử dụng giao thức SSL/ TLS để mã hóa và xác thực. HTTPS được chỉ định bởi RFC 2818 (tháng 5 năm 2000) và sử dụng cổng 443 theo mặc định thay vì cổng 80 của HTTP. HTTPS bổ sung mã hóa, xác thực và tính toàn vẹn cho giao thức HTTP:

- Mã hóa: Bởi vì HTTP ban đầu được thiết kế như một giao thức văn bản rõ ràng, nó rất dễ bị nghe trộm và tấn công trung gian. Bằng cách bao gồm mã hóa SSL/TLS, HTTPS ngăn không cho bên thứ ba chặn và đọc dữ liệu được gửi qua internet. Thông qua mật mã khóa công

CHƯƠNG 5. PHƯƠNG PHÁP GIAO TIẾP NGƯỜI DÙNG

khai và bắt tay SSL/TLS , một phiên giao tiếp được mã hóa có thể được thiết lập an toàn giữa hai bên chưa từng gặp trực tiếp (ví dụ: máy chủ web và trình duyệt) thông qua việc tạo khóa bí mật được chia sẻ.

- Xác thực: Không giống như HTTP, HTTPS bao gồm xác thực mạnh mẽ thông qua giao thức SSL/TLS. Chứng chỉ SSL/TLS của trang web bao gồm khóa công khai mà trình duyệt web có thể sử dụng để xác nhận rằng các tài liệu do máy chủ gửi (chẳng hạn như các trang HTML) đã được ký điện tử bởi ai đó sở hữu khóa cá nhân tương ứng . Nếu chứng chỉ của máy chủ đã được ký bởi tổ chức phát hành chứng chỉ đáng tin cậy công khai(CA), chẳng hạn như SSL.com, trình duyệt sẽ chấp nhận rằng mọi thông tin nhận dạng có trong chứng chỉ đã được xác thực bởi bên thứ ba đáng tin cậy. Các trang web HTTPS cũng có thể được định cấu hình để xác thực lẫn nhau, trong đó trình duyệt web trình bày một chứng chỉ ứng dụng khách xác định người dùng. Xác thực lẫn nhau rất hữu ích cho các tình huống như làm việc từ xa, nơi mong muốn bao gồm xác thực đa yếu tố, giảm nguy cơ lừa đảo hoặc các cuộc tấn công khác liên quan đến đánh cắp thông tin xác thực. Để biết thêm thông tin về cách định cấu hình chứng chỉ ứng dụng khách trong trình duyệt web, vui lòng đọc hướng dẫn này.
- Tính toàn vẹn: Mỗi tài liệu (chẳng hạn như trang web, hình ảnh hoặc tệp JavaScript) được gửi đến trình duyệt bởi máy chủ web HTTPS bao gồm chữ ký điện tử mà trình duyệt web có thể sử dụng để xác định rằng tài liệu đó không bị thay đổi bởi bên thứ ba hoặc nếu không bị hỏng khi đang vận chuyển. Máy chủ tính toán một hàm băm mật mã nội dung của tài liệu, có kèm theo chứng chỉ số của nó, mà trình duyệt có thể tính toán độc lập để chứng minh rằng tính toàn vẹn của tài liệu là nguyên vẹn.

5.1.2 Phương thức GET và POST

GET và POST là hai phương thức của giao thức HTTP, đều là gửi dữ liệu về server xử lí sau khi người dùng nhập thông tin vào form và thực hiện submit. Trước khi gửi thông tin, nó sẽ được mã hóa bằng cách sử dụng một giản đồ gọi là url encoding. Giản đồ này là các cặp name/value được kết hợp với các kí hiệu và các kí hiệu khác nhau được ngăn cách bởi dấu &.

Phương thức GET gửi thông tin người dùng đã được mã hóa thêm vào trên yêu cầu trang. GET lộ thông tin trên đường dẫn URL. Bằng thông của nó chỉ khoảng 1024 kí tự vì vậy GET hạn chế về số kí tự được gửi đi. GET

CHƯƠNG 5. PHƯƠNG PHÁP GIAO TIẾP NGƯỜI DÙNG

không thể gửi dữ liệu nhị phân, hình ảnh... Có thể cached và được đánh dấu trên trình duyệt (bookmark). Lưu trong lịch sử trình duyệt.

Phương thức POST truyền thông tin thông qua HTTP header, thông tin này được mã hóa như phương thức GET. Dữ liệu được gửi bởi phương thức POST rất bảo mật vì dữ liệu được gửi ngầm, không đưa lên URL, bằng việc sử dụng Secure HTTP, thông tin của mình có thể là an toàn. Parameters được truyền trong request body nên có thể truyền dữ liệu lớn, hạn chế tùy thuộc vào cấu hình của Server. Không cache và đánh dấu được cũng như không được lưu lại trong lịch sử trình duyệt. POST không có bất kỳ hạn chế nào về kích thước dữ liệu sẽ gửi, có thể gửi dữ liệu nhị phân, hình ảnh.

Trong lưu trữ (cache): Dữ liệu gửi bằng phương thức GET sẽ được lưu trữ lại trong query string và có thể được xem trong lịch sử trình duyệt. Ngược lại thì dữ liệu và địa chỉ URL của các request gửi bằng POST không được trình duyệt lưu lại. Tốc độ GET nhanh hơn rất nhiều so với POST về quá trình thực thi vì dữ liệu gửi đi luôn được bộ nhớ cache của trình duyệt web lưu lại, khi dùng phương thức POST thì server luôn thực thi và trả kết quả cho client, còn dùng GET thì webrowser cached sẽ kiểm tra có kết quả tương ứng đó trong cached chưa, nếu có thì trả về ngay mà không cần đưa tới server. Bookmark Đối với request gửi bằng phương thức GET người dùng có thể bookmark lại được trên trình duyệt. Ngược lại các request gửi bằng POST sẽ không thể đánh dấu được. Gửi lại form Với form gửi đi bằng phương thức GET bạn có thể gửi lại bằng cách bấm phím F5 hoặc Ctrl + R. Tuy nhiên với phương thức POST, nếu bạn muốn thực hiện việc gửi lại dữ liệu của form thì trình duyệt sẽ hiển thị một hộp thoại cảnh báo. Trở lại trang trước Trong trường hợp bạn đã gửi form dữ liệu đi rồi sau đó bấm phím Backspace để quay lại trang trước thì với phương thức GET bạn sẽ vẫn được cùng một nội dung (chứa form). Ngược lại với POST thì bạn sẽ thấy một trang trống. Bảo mật Phương thức POST bảo mật hơn GET vì dữ liệu được gửi ngầm, không xuất hiện trên URL, dữ liệu cũng không được lưu lại trong khi đó với GET thì bạn có thể hiển thị lại được các dữ liệu này. Dữ liệu Phương thức POST không giới hạn dung lượng dữ liệu gửi đi cũng như loại nhữ liệu (văn bản thông thường hay file nhị phân như upload tập tin hay hình ảnh, video...). Ngược lại, với phương thức GET dữ liệu gửi đi bị giới hạn sử dụng các ký tự chữ có trong bộ ký tự ASCII. Đồng thời dữ liệu của GET được gửi trong URL thông qua query string nên sẽ bị giới hạn bởi số lượng ký tự tối đa cho phép trong URL.

Qua những phân tích trên, GET và POST nên được sử dụng như sau:

- Khi lấy dữ liệu nên dùng GET để truy xuất và xử lí nhanh hơn.
- Khi tạo dữ liệu nên dùng POST để bảo mật dữ liệu hơn. Một ví dụ cụ thể như trong trường hợp bạn cần tạo một form để người dùng

CHƯƠNG 5. PHƯƠNG PHÁP GIAO TIẾP NGƯỜI DÙNG

upload avatar hay video thì thường sẽ sử dụng phương thức POST. Tuy nhiên khi tải về avatar hoặc video thì lại nên sử dụng phương thức GET. Ngoài ra khi cần xử lý các thông tin nhạy cảm ví dụ như email, password thì bạn cần sử dụng POST thay vì GET.

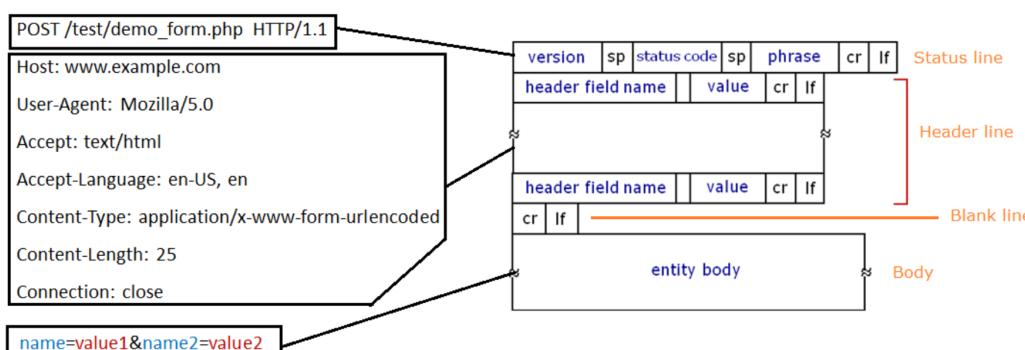
- Đối với những dữ liệu luôn được thay đổi thì nên sử dụng phương thức POST, còn dữ liệu ít thay đổi thì nên dùng phương thức GET để truy xuất và xử lý nhanh hơn.

5.1.3 Khuông dạng thông điệp yêu cầu

Khuông dạng thông điệp yêu cầu (HTTP Request) là thông tin từ máy khách đến máy chủ, yêu cầu máy chủ làm điều gì đó. Khi vào trình duyệt, gõ một địa chỉ, ngay lập tức, trình duyệt sẽ dựa vào tên miền để gửi yêu cầu truy cập đến địa chỉ IP mà tên miền này đang trả tới. Tại thời điểm này, phía máy chủ sẽ phân tích yêu cầu và sẽ gửi luồng xử lý đến vị trí lưu trữ mã nguồn. Nhiệm vụ của mã nguồn là nhận yêu cầu, phân tích yêu cầu đó và trả lại kết quả cho máy khách. Để thực hiện yêu cầu, khách hàng sử dụng các thành phần của URL (Bộ định vị tài nguyên đồng nhất), bao gồm thông tin cần thiết để truy cập tài nguyên. Các thành phần của URL giải thích các URL. Một yêu cầu HTTP được soạn đúng cách chứa các yếu tố sau:

1. Một dòng yêu cầu.
2. Một loạt các tiêu đề HTTP hoặc các trường tiêu đề.
3. Nội dung thư, nếu cần.

Mỗi tiêu đề HTTP được theo sau bởi một nguồn cấp dữ liệu dòng trả về xuống dòng (CRLF). Sau phần cuối cùng của tiêu đề HTTP, một CRLF bổ sung được sử dụng (để cung cấp một dòng trống), và sau đó bất kỳ nội dung thư nào sẽ bắt đầu.



Hình 5.1: Thông điệp yêu cầu.

CHƯƠNG 5. PHƯƠNG PHÁP GIAO TIẾP NGƯỜI DÙNG

Dòng yêu cầu

Dòng yêu cầu là dòng đầu tiên trong thông báo yêu cầu. Nó bao gồm ít nhất ba mục: Một phương pháp . Phương thức là một lệnh một từ cho máy chủ biết nó phải làm gì với tài nguyên. Ví dụ, máy chủ có thể được yêu cầu gửi tài nguyên cho máy khách. Thành phần đường dẫn của URL cho yêu cầu. Đường dẫn xác định tài nguyên trên máy chủ. Số phiên bản HTTP, hiển thị thông số kỹ thuật HTTP mà khách hàng đã cố gắng làm cho thông báo tuân thủ.

Ví dụ về dòng yêu cầu là: *POST /tes/demo_form.php HTTP/1.1*

Trong ví dụ này:

- Phương pháp là POST
- Đường dẫn là /test/demo_form.php
- Phiên bản HTTP là HTTP/1.1

Dòng yêu cầu có thể chứa một số mục bổ sung: Một chuỗi truy vấn. Điều này cung cấp một chuỗi thông tin mà tài nguyên có thể sử dụng cho một số mục đích. Nó đi theo đường dẫn và được đặt trước bởi một dấu chấm hỏi. Lực đồ và các thành phần máy chủ của URL, ngoài đường dẫn. Khi vị trí tài nguyên được chỉ định theo cách này, nó được gọi là dạng URI tuyệt đối . Đối với HTTP/1.1, biểu mẫu này được sử dụng khi một yêu cầu sẽ đi qua máy chủ proxy. Cũng đối với HTTP/1.1, nếu thành phần máy chủ của URL không được bao gồm trong dòng yêu cầu, thì nó phải được đưa vào thông báo trong tiêu đề Máy chủ.

Tiêu đề HTTP

Tiêu đề HTTP được viết trên thư để cung cấp cho người nhận thông tin về thư, người gửi và cách người gửi muốn giao tiếp với người nhận. Mỗi tiêu đề HTTP được tạo thành từ một tên và một giá trị. Các thông số kỹ thuật của giao thức HTTP xác định bộ tiêu đề HTTP tiêu chuẩn và mô tả cách sử dụng chúng một cách chính xác. Thông báo HTTP cũng có thể bao gồm các tiêu đề mở rộng, không phải là một phần của thông số kỹ thuật HTTP/1.1 hoặc HTTP/1.0. Tiêu đề HTTP cho yêu cầu của khách hàng chứa thông tin mà máy chủ có thể sử dụng để quyết định cách phản hồi yêu cầu. Một dòng trống (nghĩa là chỉ một CRLF) được đặt trong thông báo yêu cầu sau chuỗi tiêu đề HTTP, để phân chia tiêu đề khỏi nội dung thư.

Nội dung thư

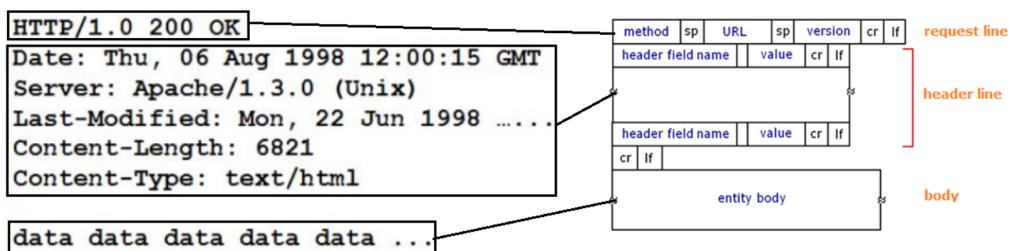
Nội dung nội dung của bất kỳ thông báo HTTP nào có thể được coi là nội dung thông báo hoặc nội dung **thực thể**. Về mặt kỹ thuật, phần thân thực thể là nội dung thực của thư. Phần nội dung thông báo chứa phần thân thực thể, có thể ở trạng thái ban đầu hoặc có thể được mã hóa theo một cách nào đó để vận chuyển, chẳng hạn như bằng cách chia nhỏ thành nhiều phần (mã hóa chuyển giao phân đoạn). Nội dung thông báo của một yêu cầu có thể được coi là một cơ quan yêu cầu để thuận tiện. Nội dung thư phù hợp với một số phương thức yêu cầu và không phù hợp với những phương thức khác. Ví dụ: một yêu cầu với phương thức POST, gửi dữ liệu đầu vào đến máy chủ, có nội dung thông báo chứa dữ liệu. Một yêu cầu với phương thức GET, yêu cầu máy chủ gửi tài nguyên, không có nội dung thư.

5.1.4 Khuông dạng thông điệp trả lời

Khuông dạng thông điệp trả lời (HTTP Response) là kết quả được thực hiện bởi một máy chủ đến một máy khách. Mục đích của phản hồi là cung cấp cho khách hàng tài nguyên mà khách hàng yêu cầu, hoặc thông báo cho khách hàng rằng hành động mà khách hàng yêu cầu đã được thực hiện; hoặc thông báo cho khách hàng rằng đã xảy ra lỗi khi xử lý yêu cầu của họ. Phản hồi HTTP chứa:

1. Một dòng trạng thái.
2. Một loạt các tiêu đề HTTP hoặc các trường tiêu đề.
3. Nội dung thư, thường là cần thiết.

Như trong một thông báo yêu cầu, mỗi tiêu đề HTTP được theo sau bởi một nguồn cấp dữ liệu dòng trả về xuống dòng (CRLF). Sau phần cuối cùng của tiêu đề HTTP, một CRLF bổ sung được sử dụng (để cung cấp một dòng trống), và sau đó nội dung thư bắt đầu.



Hình 5.2: Thông điệp trả lời.

CHƯƠNG 5. PHƯƠNG PHÁP GIAO TIẾP NGƯỜI DÙNG

Dòng trạng thái

Dòng trạng thái là dòng đầu tiên trong tin nhắn phản hồi. Nó bao gồm ba mục: Số phiên bản HTTP (hiển thị thông số kỹ thuật HTTP mà máy chủ đã cố gắng làm cho thông báo tuân thủ); Một mã trạng thái (là số có ba chữ số cho thấy kết quả của yêu cầu.) và một cụm từ lý do (hay còn gọi là văn bản trạng thái, đó là văn bản do con người có thể đọc được tóm tắt ý nghĩa của các mã trạng thái.)

Ví dụ về dòng phản hồi là: *HTTP/1.1 200 OK*

Trong ví dụ này:

- Phiên bản HTTP là HTTP/1.1
- Mã trạng thái là 200
- Cụm từ lý do là OK

Mã trạng thái và cụm từ lý do giải thích thêm về các yếu tố này của dòng trạng thái.

Tiêu đề HTTP

Các tiêu đề HTTP cho phản hồi của máy chủ chứa thông tin mà máy khách có thể sử dụng để tìm hiểu thêm về phản hồi và về máy chủ đã gửi phản hồi. Thông tin này có thể hỗ trợ máy khách hiển thị phản hồi cho người dùng, lưu trữ (hoặc bộ nhớ đệm) phản hồi để sử dụng trong tương lai và đưa ra các yêu cầu khác tới máy chủ ngay bây giờ hoặc trong tương lai. Trong trường hợp yêu cầu không thành công, tiêu đề có thể được sử dụng để cho khách hàng biết họ phải làm gì để hoàn thành yêu cầu của mình thành công. Một dòng trống (nghĩa là chỉ một CRLF) được đặt trong thông báo phản hồi sau chuỗi tiêu đề HTTP, để phân chia tiêu đề khỏi nội dung thư.

Nội dung thư

Nội dung thông báo của một phản hồi có thể được gọi để thuận tiện như một cơ quan phản hồi. Nội dung thư được sử dụng cho hầu hết các phản hồi. Các trường hợp ngoại lệ là nơi máy chủ đang phản hồi một yêu cầu máy khách đã sử dụng phương thức HEAD (yêu cầu tiêu đề nhưng không yêu cầu nội dung phản hồi) và nơi máy chủ đang sử dụng một số mã trạng thái nhất định. Để phản hồi cho một yêu cầu thành công, nội dung thông báo chứa tài nguyên được khách hàng yêu cầu hoặc một số thông tin về trạng thái của hành động được khách hàng yêu cầu. Đối với phản hồi cho một yêu cầu không thành công, nội dung thư có thể cung cấp thêm thông tin về lý do

CHƯƠNG 5. PHƯƠNG PHÁP GIAO TIẾP NGƯỜI DÙNG

gây ra lỗi hoặc về một số hành động mà khách hàng cần thực hiện để hoàn thành yêu cầu thành công.

Trạng thái và trạng thái thông điệp

Yếu tố mã trạng thái (Status-Code) là một số nguyên 3 ký tự. Ký tự đầu tiên của mã hóa trạng thái định nghĩa hạng (loại) phản hồi và hai ký tự cuối không có bất cứ vai trò phân loại nào. Có 5 giá trị của ký tự đầu tiên:

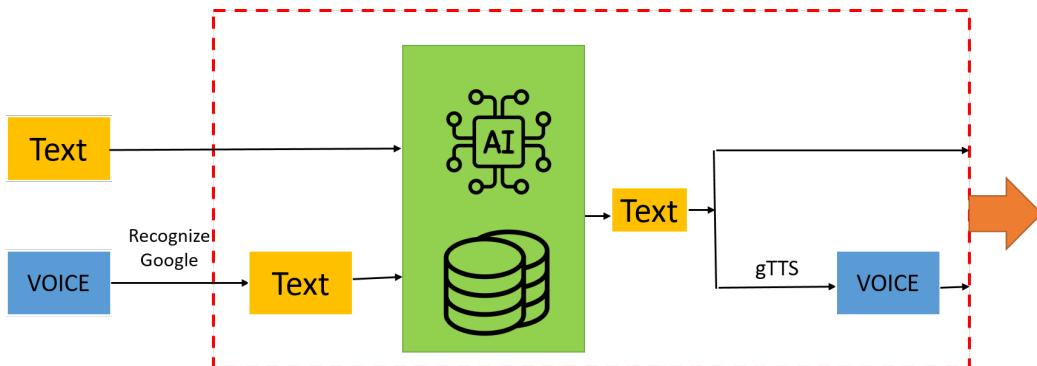
- **1xx:** Thông tin. Mã này nghĩa là yêu cầu đã được nhận và tiến trình đang tiếp tục.
- **2xx:** Thành công. Mã này nghĩa là hoạt động đã được nhận, được hiểu, và được chấp nhận một cách thành công.
- **3xx:** Chuyển hướng. Mã này nghĩa là hoạt động phải được thực hiện để hoàn thành yêu cầu.
- **4xx:** Lỗi khách hàng (Client). Mã này nghĩa là yêu cầu chứa cú pháp không chính xác hoặc không được thực hiện.
- **5xx:** Lỗi máy chủ (Server). Mã này nghĩa là máy chủ thất bại với việc thực hiện một yêu cầu nhìn như có vẻ khả thi.

5.2 Giao tiếp người dùng thông qua giao diện web

Trong đề tài này sử dụng ngôn ngữ HTML, CSS, Javascript và Python để thiết kế font-end và back-end cho trang web. HTML (Hypertext Markup Language) là ngôn ngữ đánh dấu siêu văn bản, sử dụng thẻ mở và thẻ đóng để đánh dấu cho từng đối tượng của siêu văn bản. Nhưng nếu chỉ có HTML thôi thì trang web sẽ rất thô sơ và nhảm chán, do đó cần có CSS (Cascading Style Sheet) để tạo hiệu ứng và phong cách cho các đối tượng trong HTML. Việc kết hợp thêm Javascript sẽ giúp trang web có thêm phần động và Python nằm ở phía server sẽ giúp ta xử lý các thông tin người dùng cung cấp thông qua hộp chat trên web và trả về kết quả cho người dùng cũng qua hộp chat đó. Chatbot này có hai chức năng giao tiếp theo yêu cầu bằng văn bản hoặc theo yêu cầu bằng giọng nói. Theo đó, với chức năng văn bản, người dùng chỉ cần nhập vào ô tin nhắn sau đó nhấn Enter, yêu cầu sẽ được gửi về hệ thống và xử lý. Chức năng giọng nói tương tự khi người dùng nhấn vào nút Record. Để tránh trường hợp lỗi, khi người dùng đã nhấn nút Record, lập tức nút Record sẽ bị vô hiệu hoá đến khi hệ thống trả kết quả về.

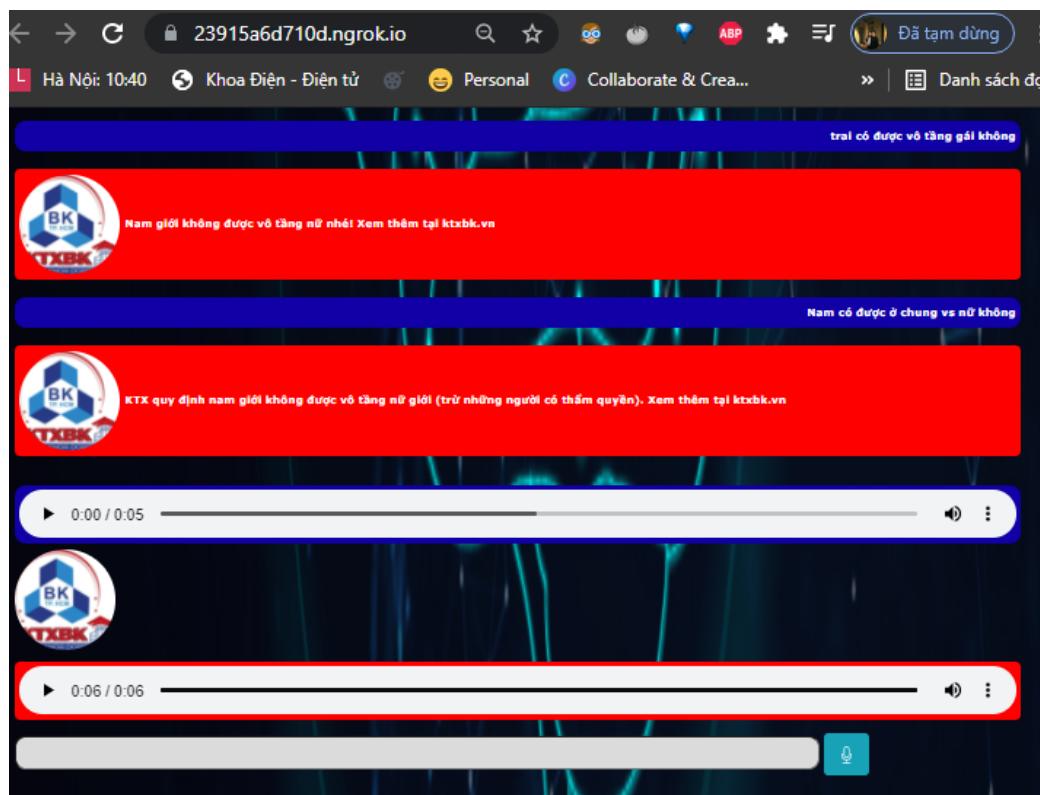
CHƯƠNG 5. PHƯƠNG PHÁP GIAO TIẾP NGƯỜI DÙNG

Khi chuyển từ giọng nói sang văn bản, công cụ Recognize Google được sử dụng và khi chuyển văn bản sang lại giọng nói, công cụ gTTS được sử dụng. Các công cụ này đều hỗ trợ trong Python và do Google API cung cấp.



Hình 5.3: Quá trình xử lý giao tiếp bằng văn bản hoặc giọng nói

Kết quả trả về khi hệ thống được yêu cầu bằng cả văn bản và giọng nói.



Hình 5.4: Giao tiếp chatbot

CHƯƠNG 5. PHƯƠNG PHÁP GIAO TIẾP NGƯỜI DÙNG

Khi đó, về phía Server, quá trình xử lý sẽ được diễn ra như hình dưới đây:

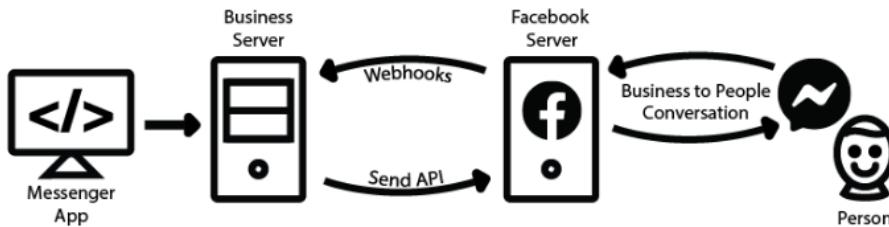
```
Connections      ttl     open    rt1     rt5     p50     p90
recording started@0al GPU (device: rflow/compiler/jit/xla_gpu_device.cc:99) Not creating XLA devices, tf_xla_enable_xla_devices not set@/task:0/device:GPU:0 with 1366 M
recording stopped
Robot: I'm listening
you: làm sao được ở chung với nữ giới không?
0.7368054764536885 ****
rule 0.7436626851558085 0.9672786
intent: 2 0.999515
intent: 1 0.0017616899
intent: 0 0.0005981161
intent: 4 0.0004071407
Robot: KTX quy định nam giới không được vô tăng nữ giới (trừ những người có thẩm quyền). Xem thêm tại ktxb.vn
127.0.0.1 - - [16/Jun/2021 19:06:57] "GET /voicebot HTTP/1.1" 200
127.0.0.1 - - [16/Jun/2021 19:06:57] "GET /static/question.wav?u=1623845217.353786 HTTP/1.1" 206
127.0.0.1 - - [16/Jun/2021 19:06:57] "GET /static/answer.mp3?u=1623845217.353786style=%22width=70%;%22 HTTP/1.1" 206 -
```

Hình 5.5: Quá trình xử lý ở server

5.3 Giao tiếp người dùng thông qua giao diện messenger facebook

Facebook là mạng xã hội phổ biến nhất trên thế giới với 2,85 tỷ người dùng hoạt động hàng tháng (tính đến ngày 31 tháng 3 năm 2021). Vì vậy, ứng dụng sẽ cực kỳ dễ tiếp cận nếu triển khai ứng dụng chatbot trên Facebook messenger. Có 3 bước chính để triển khai một chatbot trên Facebook Messenger:

- Tạo một máy chủ nghe tin nhắn từ Facebook (sử dụng flask)
- Xác định một chức năng để gửi lại tin nhắn cho người dùng (sử dụng các yêu cầu)
- Chuyển tiếp kết nối https đến máy chủ web



Hình 5.6: Quá trình giao tiếp với server Facebook

Bước đầu tiên là tạo một máy chủ http để lắng nghe tin nhắn do Facebook gửi, nhận phản hồi cho tin nhắn đó và cuối cùng gửi phản hồi lại cho người dùng trên Facebook. Flask đã được sử dụng để tạo máy chủ này. Tạo một ứng

CHƯƠNG 5. PHƯƠNG PHÁP GIAO TIẾP NGƯỜI DÙNG

dụng bình lắng nghe các tin nhắn được gửi đến <https://kien.nkhoa.email/webhook>. Khi tin nhắn được gửi trên Facebook, chúng sẽ đến dưới dạng yêu cầu http tới URL này. Hàm nghe () xử lý các yêu cầu http này và kiểm tra xem chúng có chứa thông báo Facebook hợp lệ hay không. Nếu tin nhắn hợp lệ, hàm get_bot_response () sẽ được gọi và phản hồi được gửi trả lại Facebook Messenger. Ngoài ra, sẽ phải tạo một hàm có tên verify_webhook () để xử lý xác thực ban đầu giữa Facebook và ứng dụng. Các biến VERIFY_TOKEN và PAGE_ACCESS_TOKEN chưa được xác định - vì vậy sẽ phải đặt các biến này. VERIFY_TOKEN có thể là một chuỗi bất kỳ, nó được sử dụng như một loại mật khẩu để cho Facebook biết rằng máy chủ này muốn nhận tin nhắn. Nên sử dụng một chuỗi dài ngẫu nhiên. Tiếp theo, một chức năng gửi phản hồi trả lại Facebook Messenger đã được viết bằng cách sử dụng thư viện python được gọi là yêu cầu. Cụ thể, khi sử dụng phương thức post(), phương thức này tạo ra một yêu cầu HTTP POST. Yêu cầu POST là cách thông thường để gửi thông tin đến máy chủ. Sau đó, thiết lập Facebook Messenger. Tạo một trang facebook và ứng dụng facebook. Thêm Webhook vào ứng dụng. Tìm nạp ID ứng dụng và cập nhật nó trong tập lệnh. Thêm Messenger vào ứng dụng và tạo mã thông báo cho trang phải sử dụng để trò chuyện và chọn các sự kiện để đăng ký trang. Di tới cài đặt cho Messenger, cuộn xuống mục "Tạo mã thông báo" và nhấp vào liên kết để tạo trang mới cho ứng dụng.

Khi đã tạo một trang, hãy quay lại cài đặt. Tạo mã thông báo và chọn trang này từ menu thả xuống. Sao chép Mã thông báo truy cập trang vào trình giữ chỗ cho PAGE_ACCESS_TOKEN. Cuối cùng, cần đăng ký webhook của mình trên trang Facebook dành cho nhà phát triển. Chuyển đến tab Messenger trong Sản phẩm một lần nữa và cuộn xuống Webhooks, nhấp vào Thiết lập Webhooks, nhập URL gọi lại vào URL. Điều quan trọng là ứng dụng flask đang chạy tại thời điểm này, vì verify_token () sẽ được gọi ở bước tiếp theo. Trong trường Xác minh Mã thông báo, hãy đặt giá trị cho biến. Trong Trường Đăng ký, hãy đảm bảo rằng các tin nhắn và message_postbacks được đánh dấu. Nhập vào Xác minh và Lưu để thực hiện xác thực.

Khi một người gửi tin nhắn đến doanh nghiệp trên Messenger và miễn là Trang này sử dụng ứng dụng để tự động hóa một phần hoặc hoàn toàn các cuộc trò chuyện, điều sau sẽ xảy ra. Máy chủ Facebook gửi webhook đến URL của máy chủ doanh nghiệp, nơi lưu trữ ứng dụng nhận tin. Bằng cách sử dụng API Gửi, ứng dụng có thể trả lời người đó trên Messenger. Bằng cách này, các nhà phát triển có thể tạo các cuộc trò chuyện có hướng dẫn để dẫn dắt mọi người thông qua quy trình tự động hoặc xây dựng một ứng dụng để làm cầu nối giữa các đại lý và sự hiện diện của doanh nghiệp trên Messenger.

CHƯƠNG 5. PHƯƠNG PHÁP GIAO TIẾP NGƯỜI DÙNG

Webhooks

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Callback URL

<https://kien.nkhoa.email/webhook>

Verify Token

.....

Validation requests and Webhook notifications for this object will be sent to this URL.

Token that Facebook will echo back to you as part of callback URL verification.

[Edit Callback URL](#)

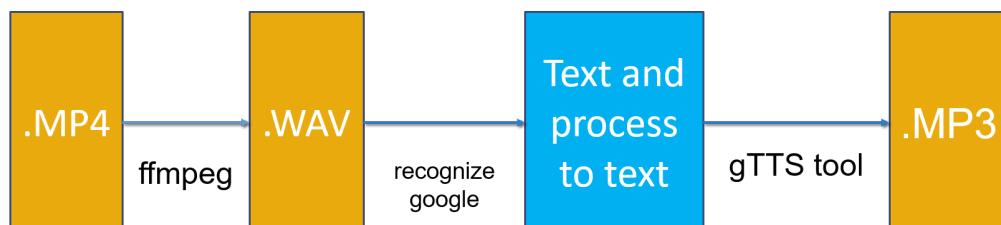
[Show Recent Errors](#)

Pages ↑	Webhooks	
 Test 108188354790151	2 Fields messages, messaging_postbacks	Edit

Hình 5.7: Cài đặt Webhook.

Trong giao tiếp âm thanh, file âm thanh thu được từ Facebook Messenger có dạng .mp4 qua 1 url. Bởi vì công cụ nhận dạng và chuyển giọng nói qua văn bản (recognize_google) không hỗ trợ xử lý file âm thanh định dạng .mp4 nên phải chuyển định dạng của file câu hỏi nhận vào từ .mp4 sang .wav bằng công cụ FFmpeg, là một giải pháp hoàn chỉnh, đa nền tảng để ghi, chuyển đổi và phát trực tuyến âm thanh và video.

Quá trình sau đó giống như quá trình xử lý tin nhắn âm thanh nhận vào từ hộp chat trên trang web.



Hình 5.8: Quá trình lấy và gửi câu trả lời âm thanh.

Trong chương này đã trình bày được cách thức giao tiếp người dùng qua giao diện Web cũng như qua Facebook Messenger, kết quả và định hướng phát triển trong tương lai của đề tài sẽ được trình bày trong chương sau.

Chương 6

KẾT QUẢ ĐỀ TÀI VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

Ở chương này, ta tiến hành nghiệm thu hệ thống, từ đó hiệu chỉnh các thông số hay cơ sở dữ liệu cho phù hợp nhằm nâng cao độ tin cậy cho hệ thống. Đồng thời ta định hướng phát triển đề tài cho phù hợp với thực tế, với nhu cầu người sử dụng cũng như xu hướng phát triển của công nghệ.

6.1 Kết quả đề tài

Với khoảng 80 bộ câu hỏi/trả lời đã được chuẩn bị, hãy xét kết quả xử lý qua từ 2 dạng đầu vào hệ thống.



Hình 6.1: Kết quả giao diện trên website.

CHƯƠNG 6. KẾT QUẢ ĐỀ TÀI VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

6.1.1 Đối với văn bản

Giả sử ta có 70 câu hỏi ngẫu nhiên, phân bố trên các chủ đề. Khi đưa vào hệ thống, ta sẽ cho ra kết quả của hệ thống trả về.

Bảng 6.1: Kết quả xử lý với đầu vào văn bản.

STT	Câu hỏi	Câu trả lời	Kết quả	Chủ đề
1	xin chào	Chào bạn ! Mình là Chatbot hỗ trợ tư vấn! Mình có thể giúp gì cho bạn ?	1	greet
2	Hôm nay trời đẹp không?	Hiện tại Bot chưa hiểu yêu cầu của bạn! Bạn vui lòng thử lại nhé!	1	Noanswer
3	khi nào được ăn cơm	Sinh viên Bách Khoa và những người có nhu cầu thuê	0	rule
4	bye	Tạm biệt bạn ! Nếu có gì thắc mắc, bạn cứ tìm đến Chatbot nhé !	1	greet
5	cảm ơn nhiều	Chúng tôi luôn sẵn sàng giải đáp thắc mắc cho cậu !	1	greet
6	giá tiền phòng dịch vụ 4 người	Giá tiền phòng dịch vụ 4 người là 2,000,000 VND/tháng	1	fee
7	giá tiền phòng 6 người dịch vụ	Giá tiền phòng dịch vụ 6 người là 1,500,000 VND/tháng	1	fee
8	còn giá tiền phòng thường	Giá tiền phòng thường 8 người là 500,000 VND/tháng	1	fee
9	giá tiền phòng thường 6 người	Giá tiền phòng thường 6 người là 800,000 VND/tháng	1	fee

CHƯƠNG 6. KẾT QUẢ ĐỀ TÀI VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

10	giá phòng rẻ nhất ở ktx	KTX đã hết phòng thường 8 người bạn nhé.	0	fee
11	tiền phòng thường 4 người	Giá tiền phòng thường 4 người là 1,200,000 VND/tháng	1	fee
12	thuê tủ lạnh 1 tháng bao nhiêu tiền	Giá thuê tủ lạnh là 200,000 VND/phòng/tháng	1	fee
13	giá tiền dịch vụ thuê tủ lạnh	Giá thuê tủ lạnh là 200,000 VND/phòng/tháng	1	fee
14	tiền thuê máy giặt	Giá thuê máy giặt là 200,000 VND/phòng/tháng	1	fee
15	giá tiền dịch vụ thuê tủ lạnh	Giá thuê tủ lạnh là 200,000 VND/phòng/tháng	1	fee
16	phí gửi xe hành tháng	Giá gửi xe máy là 70,000 VND/tháng	1	fee
17	tiền gửi xe đạp	Giá gửi xe máy là 70,000 VND/tháng	0	fee
18	phí gửi xe đạp	Giá gửi xe đạp là 30,000 VND/tháng	1	fee
19	Tiền sửa chữa đồ hư hỏng là bao nhiêu?	Chi phí sửa chữa, thay thế vật tư tính theo thời gian, vui lòng truy cập ktxbk.vn để biết thêm chi tiết.	1	fee
20	giá tiền thuê lò nướng	KTX không cho thuê cũng như sử dụng lò nướng, lò vi sóng các loại trong phòng ở để đảm bảo an toàn cháy nổ!	1	greet

CHƯƠNG 6. KẾT QUẢ ĐỀ TÀI VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

21	giám đốc ktx là ai?	Ông: Trần Tuấn Phúc là Giám đốc Trung Tâm dịch vụ KTX Bách Khoa. Bạn có thể gặp người này tại Phòng Giám đốc.	1	staff
22	trưởng phòng hành chính là ai	Bà: Nguyễn Thị Ái Trâm là Kế toán trưởng - Trưởng phòng Tài chính kế hoạch. Bạn có thể gặp người này tại Văn phòng.	0	staff
23	trợ lý giám đốc là ai ?	Ông: Nguyễn Văn Đạt là Trợ lý Giám đốc. Bạn có thể gặp người này tại Văn phòng	1	staff
24	Ai là kế toán của ktx?	Bà: Nguyễn Ngô Kiêm Ái là Kế toán . Bạn có thể gặp người này tại Văn phòng.	1	staff
25	trưởng phòng tài chính là ai	Bà: Nguyễn Thị Ái Trâm là Kế toán trưởng - Trưởng phòng Tài chính kế hoạch. Bạn có thể gặp người này tại Văn phòng.	1	staff
26	ai là thủ quỹ ktx	Bà: Bùi Thị Anh Thư là Thủ quỹ .Bạn có thể gặp người này tại Văn phòng.	1	staff
27	Thư ký văn phòng là ai	Bà: Nguyễn Thị Kim Bích là Thư ký văn phòng. Bạn có thể gặp người này tại Văn phòng.	1	

CHƯƠNG 6. KẾT QUẢ ĐỀ TÀI VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

28	chị Uyên là ai?	Bà: Vũ Dõ Uyên là Trưởng phòng Hành chính. Bạn có thể gặp người này tại Văn phòng.	1	staff
29	Trưởng Ban QLTN là ai?	Ông: Đoàn Minh Vững là Phó Giám đốc - Trưởng Ban QLTN. Bạn có thể gặp người này tại Phòng Phó giám đốc ktx.	1	staff
30	ai là phó giám đốc ktx?	Ông: Đoàn Minh Vững là Phó Giám đốc - Trưởng Ban QLTN. Bạn có thể gặp người này tại Phòng Phó giám đốc ktx.	1	staff
31	chị Oanh là ai?	Bà: Đặng Thị Thu Oanh là Thư ký văn phòng. Bạn có thể gặp người này tại Văn phòng.	1	staff
32	xin chào	Chào bạn ! Mình là Chatbot hỗ trợ tư vấn! Mình có thể giúp gì cho bạn ?	1	staff
33	ai là người phụ trách nội chính?	Ông: Lê Thành Phương là Phó Ban quản lý. Bạn có thể gặp người này tại Văn phòng.	1	staff
34	Ai là kỹ thuật viên sửa chữa ở ktx?	Ông: Vũ Đức Phương là Kỹ thuật viên sửa chữa. Bạn có thể gặp người này tại Văn phòng.	1	staff
35	ai là người phụ trách kỹ thuật ở ktx	Sinh viên Bách Khoa và những người có nhu cầu thuê	0	staff

CHƯƠNG 6. KẾT QUẢ ĐỀ TÀI VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

36	chủ tịch công đoàn ở ktx là ai	Ông: Vũ Đức Phương là Chủ tịch công đoàn và Kỹ thuật viên sửa chữa .Bạn có thể gặp người này tại Văn phòng.	1	staff
37	bị thương ở ktx thì gặp ai?	Hiện tại Bot chưa hiểu yêu cầu của bạn! Bạn vui lòng thử lại nhé!	0	staff
38	ai lo về y tế ở ktx	Bà: Trịnh Nguyễn Hoàng Anh là Quản lý tầng 2 và nhân viên y tế. Bạn có thể gặp người này tại phòng y tế.	1	staff
39	còn ai là y tá ở ktx không?	Bà: Trần Nguyễn Như Ngọc là y tá. Bạn có thể gặp người này tại phòng y tế.	1	staff
40	Cần dọn dẹp thì liên hệ ai?	Bà: Trần Nguyễn Như Ngọc là NV Phục vụ. Bạn có thể liên hệ người này qua Ban quản lý tòa nhà.	1	staff
41	nhân viên bảo vệ ở ktx là ai?	Ông: Nguyễn Phú Đức là đội trưởng đội bảo vệ. Bạn có thể gặp người này tại phòng bảo vệ.	1	staff
42	còn bảo vệ nào khác không?	CÔng: Ông: Trần Văn Trí là bảo vệ. Bạn có thể gặp người này tại phòng bảo vệ.	1	staff
43	những ai được ở trong ktx?	Sinh viên Bách Khoa và những người có nhu cầu thuê.	1	rule
44	ai được ưu tiên đăng ký ở ktx?	Sinh viên năm nhất sẽ được ưu tiên hơn cả, ngoài ra các đợt ưu tiên khác nhau sẽ có những đối tượng ưu tiên khác nhau.	1	rule

CHƯƠNG 6. KẾT QUẢ ĐỀ TÀI VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

45	có được hút thuốc trong ktx không?	Nghiêm cấm hút thuốc bên trong toà nhà. KTX có khu vực hút thuốc là sân sau.	1	rule
46	Dược hút thuốc là hay vape trong ktx không?	Hiện tại Bot chưa hiểu yêu cầu của bạn! Bạn vui lòng thử lại nhé!	0	rule
47	có được nấu cơm trong ktx không?	Không nấu nướng, tàng trữ dụng cụ để chế biến và làm chín thức ăn trong phòng ở.	1	rule
48	có gửi ô tô trong ktx dc không?	Xin lỗi, KTX chưa có dịch vụ gửi xe ô tô cho sinh viên	1	rule
49	điều kiện để được ở tiếp ở ktx là gì?	Hiện tại Bot chưa hiểu yêu cầu của bạn! Bạn vui lòng thử lại nhé!	0	rule
50	Sinh viên tốt nghiệp có thẻ được ở thêm ở ktx được không?	Gia hạn sinh viên đang ở, không bị cảnh cáo, đủ tổng điểm lưu trú sẽ chọn phòng theo từng đợt ưu tiên.	1	rule
51	Có thẻ tham gia các clb ở ktx được không?	Gia hạn sinh viên đang ở, không bị cảnh cáo, đủ tổng điểm lưu trú sẽ chọn phòng theo từng đợt ưu tiên.	0	rule
52	có được sinh hoạt câu lạc bộ ở ktx được không?	Bạn vui lòng xem thêm tại mục 'các câu lạc bộ' trên trang web: ktxbk.vn	1	rule
52	KTX có nhận hộ hàng từ shipper được không?	KTX không nhận hộ hay giữ hộ hàng mua online nhé!	1	rule

CHƯƠNG 6. KẾT QUẢ ĐỀ TÀI VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

53	mua hàng online từ ktx được không?	KTX không nhận hộ hay giữ hộ hàng mua online nhé!	0	rule
54	KTX cấm làm điều gì ở ktx?	KTX ở 497 Hoà Hảo, phường 7, Quận 10, tpHCM. Còn với mình thì KTX ở trong tym bạn nhé!	0	rule
55	không được làm gì ở ktx?	Bạn có thể đọc thêm tại mục 'quy định xử lý kỷ luật' trên trang web: ktxbk.vn.	1	rule
56	được ăn nhậu trong KTX không?	Không được phép sử dụng các thức uống có cồn như bia rượu trong ktx nhé.	1	rule
57	Ăn ở đâu trong ktx?	Phòng 101 là canteen	1	room
58	có tập võ ở đâu?	Hiện tại Bot chưa hiểu yêu cầu của bạn! Bạn vui lòng thử lại nhé!	0	room
59	tập gym ở đâu?	Phòng 102 là phòng gym.	1	room
60	các clb sinh hoạt ở đâu?	Phòng 104 là Phòng tự học chung - sinh hoạt clb.	1	room
61	Gặp giám đốc ktx ở đâu?	CPhòng 105 là Phòng Giám đốc KTX.	1	room
62	Đến đâu để đóng tiền điện?	Giá thuê máy giặt là 200,000 VND/phòng/tháng	0	room
63	Đóng tiền ở chỗ nào?	CKTX nằm sát siêu thị CoopMart Lý Thường Kiệt, gần sân banh Thống Nhất.	0	room
64	Phòng tài chính ở đâu?	Phòng 108 là Phòng Tài chính kế hoạch.	1	room

CHƯƠNG 6. KẾT QUẢ ĐỀ TÀI VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

65	phòng nào là văn phòng?	Phòng 109 là Văn phòng.	1	room
66	Trong ktx khám bệnh ở đâu?	Phòng 110 là phòng y tế.	1	room
67	KTX còn phòng trống không?	KTX còn tổng cộng 30 phòng trống, không biết bạn quan tâm đến loại nào?	1	room
68	có những loại phòng nào ở ktx?	Có tất cả 5 loại phòng: phòng thường 4 người, phòng thường 6 người, phòng thường 8 người, phòng dịch vụ 4 người, phòng dịch vụ 6 người.	1	room
69	Xin địa chỉ của ktx với ad?	Hiện tại Bot chưa hiểu yêu cầu của bạn! Bạn vui lòng thử lại nhé!	0	room
70	KTX ở đâu	KTX ở 497 Hoà Hảo, phường 7, Quận 10, tpHCM.	1	room

Với nhiều lần thử nghiệm, thấy rằng về cơ bản, tỷ lệ số câu trả lời đúng trên tổng số câu trả lời cơ bản là trên 70% và thấy được mức độ biến động về mức chính xác giữa các lớp so với giá trị trung bình là tương đối nhỏ

6.1.2 Đối với âm thanh

Đối với đầu vào âm thanh, sự chính xác được nhận thấy kém hơn so với đầu vào bằng văn bản do: Ảnh hưởng tạp âm từ môi trường, đôi khi giọng nói bạn không đủ to hay phát âm không rõ ràng và do độ trễ của hệ thống làm đầu ghi âm thanh trễ hơn so với lúc người bắt đầu nói, chất lượng của thiết bị thu âm,... Sau đây là kết quả khảo sát:

CHƯƠNG 6. KẾT QUẢ ĐỀ TÀI VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

Bảng 6.2: Kết quả xử lý với đầu vào giọng nói.

STT	Câu hỏi đúng	Câu hỏi nghe được	Kết quả phản hồi	Thời gian phản hồi (s)
1	Giám đốc ktx là ai	Chào bạn ! Mình là Chatbot hỗ trợ tư vấn! Mình có thể giúp gì cho bạn ?	1	greet
2	Ký túc xá ở đâu	ký túc xá ở đâu	1	0.5
3	xin chào	xin chào	1	1.5
4	thank you so much	thank you so much	1	2.5
5	giá tiền thuê phòng 4 người	giá tiền thuê phòng 4 người	1	0.5
6	giá tiền thuê phòng dịch vụ 6 người	giá tiêm phòng dịch vụ 6 người	0	1.5
7	được hút thuốc trong ký túc xá không	được hút thuốc trong ký túc xá không	1	2
8	chị oanh là ai	chị oanh là ai	1	1.5
9	cho xin địa chỉ ký túc xá với	cho xin địa chỉ ký túc xá với	0	3.5
10	ký túc xá có nhận hộ hàng từ shipper không	ký túc xá có nhận hộ hàng từ shipper không	1	2

Như vậy, dựa vào bảng trên, tỷ lệ giữa số câu nghe đúng trên tổng số câu là rất cao (9/10). Con số này cho thấy việc nhận đầu vào của hệ thống khá tốt và ổn định như đầu vào là văn bản.

6.2 Ưu điểm và hạn chế của đề tài

Một trong những ưu điểm của đề tài đó là tính ứng dụng cao và tính giản đơn trong cấu trúc, dễ dàng tiếp cận và sử dụng.

Về mặt đơn giản về cấu trúc và tính đa dạng tiếp cận thông qua các công cụ xây dựng hệ thống. Ứng dụng được xây dựng dựa trên ngôn ngữ Python, là ngôn ngữ vừa đơn giản lại vừa phổ biến. Ngoài ra mô hình máy chủ giúp nhiều người có thể truy cập hệ thống cùng lúc cũng như kết nối đến mạng xã hội Facebook. Cùng với việc sử dụng MySQL để lưu trữ và xử lý dữ liệu, ta có thể dễ dàng thay đổi câu trả lời khi có sự thay đổi thật.

Tuy nhiên đề tài còn có một vài điểm hạn chế về số lượng thông tin chưa bao quát hết so với khối lượng thông tin thật sự. Giao diện Web còn thô sơ, chưa bắt mắt. Ngoài ra hệ thống chưa thật sự giám sát được số lượng người truy cập cùng lúc để xử lý quá tải nếu có. Về tính chính xác về mặt thông tin, hệ thống đôi khi còn đưa ra một số câu trả lời còn nhập nhằng, chưa rõ ràng, đúng với ý của người hỏi. Thời gian phản hồi chưa nhanh, phụ thuộc thêm vào độ dài câu trả lời.



Hình 6.2: Hệ thống quá tải khi phải xử lý nhiều yêu cầu cùng lúc.

6.3 Định hướng phát triển đề tài

Định hướng phát triển chính của đề tài là cải thiện tính chính xác cũng như mở rộng thêm dữ liệu hệ thống. Đồng thời cũng cần phải tăng thêm tính bảo mật của hệ thống về cả cơ sở dữ liệu và mạng máy tính. Ngoài ra một số tính năng đặc biệt có thể được phát triển thêm cho hệ thống:

- Phát triển tính năng chuyển đổi đa ngôn ngữ trong thời gian thực.
- Triển khai trên nhiều nền tảng mạng xã hội, công cụ nhắn tin khác như: Telegram, Viber, Skyper, Zalo,...
- Phát triển thêm tính năng tích hợp với công cụ tìm kiếm như google.

Tài liệu tham khảo

- [1] L.C. Evans, *Partial differential equations*, Graduate Studies in Mathematics Vol. 19, Amer. Math. Soc., Providence, Rhode Island, 1998.
- [2] Simon Haykin, (1999). *Neural Network: A Comprehensive Foundation*. Pearson Education, Inc.
- [3] Phan Thị Tươi, (2012). *Xử lý ngôn ngữ tự nhiên*. Tp.HCM: NXB Đại học quốc gia TPHCM.
- [4] Oracle Corporation (2020). MySQL 8.0 Reference Manual. Online 8/6/2021, from <https://dev.mysql.com/doc/refman/8.0/en/>
- [5] Nguyễn Phú Cường - Hoàng Ngọc Vinh (1/2021). *Hệ thống Chatbot hỗ trợ tư vấn sinh viên*
- [6] Amar Budhiraja (12/2016). *Dropout in (Deep) Machine learning*. Online 8/6/2021, from <https://medium.com/@amarbudhiraja/https-mediumcom-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machinelearning-74334da4bfc5>
- [7] *HTML and CSS, JavaScript, Python Tutorial*. Online 10/5/2021, from <https://www.w3schools.com/>
- [8] Viet-Anh Nguyen (7/201), *Các hàm kích hoạt (activation function) trong neural network*. Online 10/6/2021, form <https://aicurious.io/posts/2019-09-23-cac-ham-kich-hoat-activation-function-trong-neural-networks/>
- [9] François Chollet (December 22, 2017), *Deep Learning with Python 1st Edition*. Manning Publications.
- [10] Jason Brownlee (January 18, 2021), *How to Choose an Activation Function for Deep Learning*. Online 10/6/2021, form: <https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/>

TÀI LIỆU THAM KHẢO

- [11] Jason Brownlee (March 16, 2016). *Supervised and Unsupervised Machine Learning Algorithms*. Online 10/6/2021, form: <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>
- [12] Cuong Tran (Aug 25, 2019), *Thuật Toán Tối Ưu Gradient Descent*. Online 10/6/2021, form: <https://medium.com/@vicohub/thuật-toán-tối-ưu-gradient-descent-21a0a397928>
- [13] Do Minh Hai (4/2017). *Mạng nơ-ron nhân tạo - Neural Networks*. Online 10/6/2021, form: <https://dominhhai.github.io/vi/2018/04/nn-intro/>
- [14] Léon Bottou (30 September 2010). *Large-Scale Machine Learning with Stochastic Gradient Descent*. NEC Labs AmericaPrincetonUSA.
- [15] Sebastian Ruder *An overview of gradient descent optimization algorithms*. Online 10/6/2021, form: <https://ruder.io/optimizing-gradient-descent/index.html#stochasticgradientdescent>
- [16] Online 30/6/2021, form: <https://docs.gunicorn.org/en/stable/index.html>
- [17] Đào Đăng Toàn *Chạy ứng dụng Flask với Gunicorn trên ubuntu*. Online 30/6/2021, form: <https://vimentor.com/vi/lesson/6-chay-ung-dung-flask-voi-gunicorn-tren-ubuntu>
- [18] Alan Nichol (February 28th, 2018), *Deploy Your Facebook Messenger Bot with Python*, Online 30/6/2021, form: <https://www.datacamp.com/community/tutorials/facebook-chatbot-python-deploy>
- [19] Agendaless Consulting and Contributors (Jul 26, 2021), *Supervisor's Docs*. Online 30/6/2021, form: <http://supvisord.org/>
- [20] Introduction to the Messenger Platform. Online 30/6/2021, form: <https://developers.facebook.com/docs/messenger-platform/introduction/>, Accessed on: Jul. 2, 2021.
- [21] James Marshall (December 10, 2012). *HTTP Made Really Easy*, Online 10/6/2021, form: <https://www.jmarshall.com/easy/http/#toc>
- [22] Aaron Russell (13/7/2020). *SSL document: "What is HTTPS?"*, Online 10/6/2021, form: <https://www.ssl.com/faqs/what-is-https/>
- [23] IBM® Knowledge Center, *the CICS documentation*. Online 10/6/2021, form: <https://www.ibm.com/docs/en/cics-ts/5.3?topic=concepts-http-protocol>