BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC NÔNG LÂM TP. HCM

KHOA CÔNG NGHỆ THÔNG TIN

**ĐỒ ÁN MÔN**

**NHẬP MÔN TRÍ TUỆ NHÂN TẠO**

Sinh viên thực hiện MSSV

Lê Quốc Thịnh 19130215

Nguyễn Công Phúc 19130172

Ngô Thành Đề 19130035

Nguyễn Ngọc Hiếu 19130072

Giảng viên bộ môn: Nguyễn Văn Dũ

Ngày -- tháng -- năm --

Mục lục

[I. Giới thiệu chung 2](#_Toc97302789)

[1. Giới thiệu trợ lý ảo 2](#_Toc97302790)

[2. Lý do chọn python 2](#_Toc97302791)

[3. Các chức năng của trợ lý ảo cơ bản 2](#_Toc97302792)

[II. Tổng quan các bước xây dựng 3](#_Toc97302793)

[1. Import các thư viện cần thiết 3](#_Toc97302794)

[2. Cài đặt nơ-ron 3](#_Toc97302795)

[III. Các bước thực hiện 10](#_Toc97302796)

[1. Các bước hoạt động 10](#_Toc97302797)

[2. Chuyển giọng nói thành văn bản 10](#_Toc97302798)

[3. Xử lý yêu cầu 11](#_Toc97302799)

[4. Chuyển văn bản thành giọng nói 11](#_Toc97302800)

[5. Tổng kết 11](#_Toc97302801)

# I. Giới thiệu chung

## 1. Giới thiệu trợ lý ảo

Một trợ lý ảo là một [tác tử phần mềm](https://vi.wikipedia.org/wiki/T%C3%A1c_t%E1%BB%AD_ph%E1%BA%A7n_m%E1%BB%81m) có khả năng thực hiện các tác vụ hoặc dịch vụ cho cá nhân người sử dụng.Tính tới 2017, các tính năng và lượng người sử dụng của các trợ lý ảo đang tăng lên nhanh chóng, với các sản phẩm mới luôn được cho ra thị trường. Một khảo sát trực tuyến vào tháng 5 năm 2017 cho thấy những trợ lý ảo được dùng rộng rãi nhất tại Hoa Kỳ là [Apple Siri](https://vi.wikipedia.org/wiki/Apple_Siri) (34%), [Google Assistant](https://vi.wikipedia.org/wiki/Google_Assistant) (19%), [Amazon Alexa](https://vi.wikipedia.org/wiki/Amazon_Alexa) (6%), và [Microsoft Cortana](https://vi.wikipedia.org/wiki/Microsoft_Cortana) (4%). [Trợ lý ảo M](https://vi.wikipedia.org/wiki/M_(tr%E1%BB%A3_l%C3%BD_%E1%BA%A3o)) của [Facebook](https://vi.wikipedia.org/wiki/Facebook) được dự kiến phát hành tới hàng trăm triệu người dùng trên [Facebook Messenger](https://vi.wikipedia.org/wiki/Facebook_Messenger) vào năm 2017. Apple và Google có các trợ lý ảo được cài đặt nhiều nhất trên các [điện thoại thông minh](https://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_tho%E1%BA%A1i_th%C3%B4ng_minh) và Microsoft là trên các [máy tính cá nhân](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_c%C3%A1_nh%C3%A2n) chạy hệ điều hành [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows) của hãng (trợ lý ảo Cortana của hãng được tích hợp vào hệ điều hành này cùng với các điện thoại và loa thông minh); trong khi đó, Alexa là sản phẩm đầu tiên có khả năng đặt hàng trực tuyến từ Amazon.

## 2. Lý do chọn python

Lý do chọn python: vì python đễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o). Nó hỗ trợ nhiều thư viện để xây dựng lên một machine learning.

## 3. Các chức năng của trợ lý ảo cơ bản

- Chào hỏi, giao tiếp

- Cho biết ngày giờ

- Mở ứng dụng

- Mở google và tìm kiếm

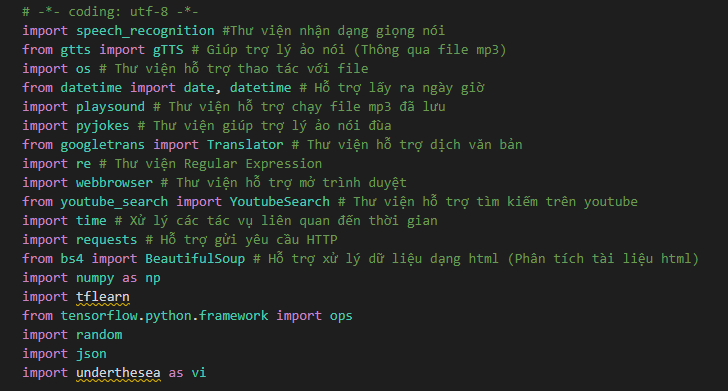
- Tìm kiếm trên youtube

- Tìm kiếm trong ổ đĩa máy tính

- Tán gẫu nói chuyện

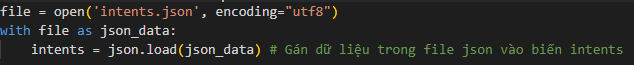
# II. Tổng quan các bước xây dựng

## 1. Import các thư viện cần thiết



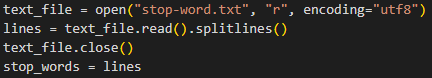
## 2. Cài đặt nơ-ron

Đọc dữ liệu trong file json (File dataset)

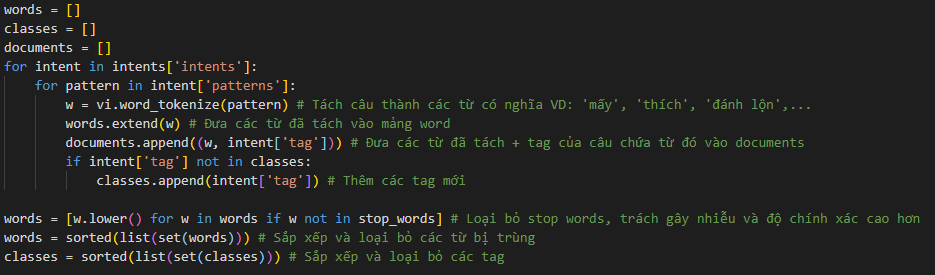


Lấy ra stop word trong file stop-word.txt

STOP WORD: hiểu đơn giản là các từ có tần số xuất hiện nhiều như à, à này, ... các từ này thường mang ít giá trị ý nghĩa và không khác nhau nhiều trong các văn bản khác nhau



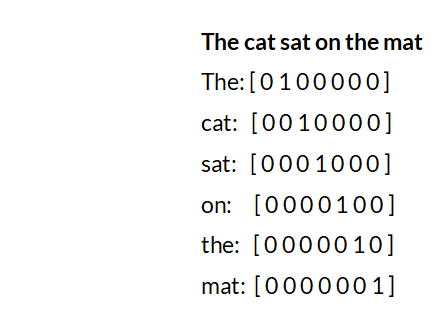
Xử lý dữ liệu train



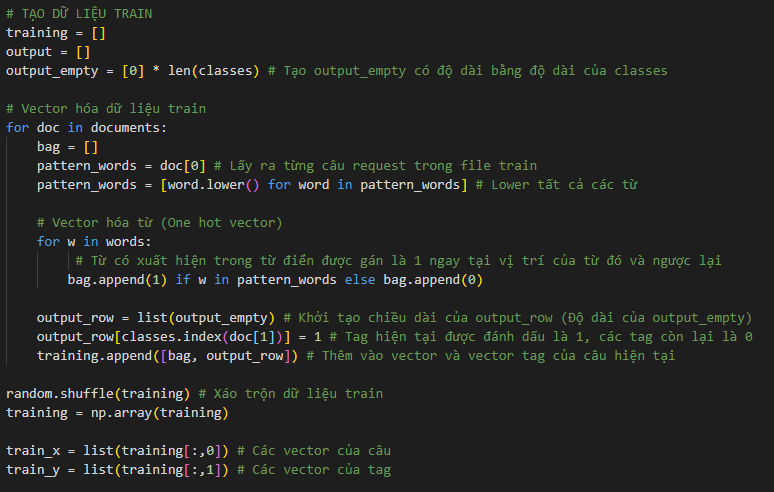
Tạo dữ liệu train

Nếu chúng ta sử dụng dự liệu dạng word thì sẽ không thể hoạt động được với tensorflow, vì vậy việc cần thiết bây giờ là chuyển dữ liệu này sang dạng tensor number

One hot vector: mỗi giá trị hạng mục sẽ được mã hóa bằng một vector nhị phân với toàn bộ các phần tử bằng 0 trừ một phần tử bằng 1 tương ứng với vị trí của giá trị hạng mục đó trong từ điển.

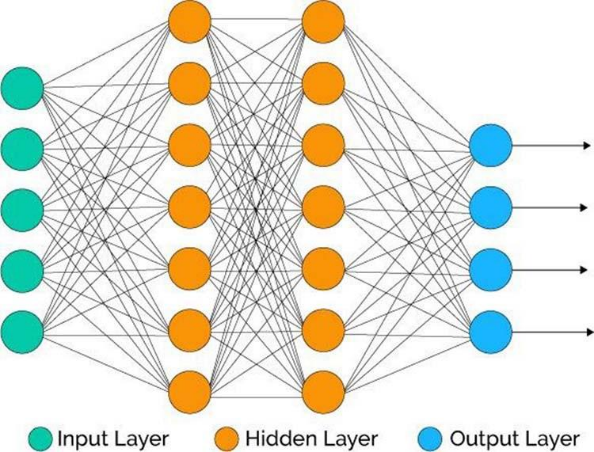


One hot vector: đầu ra là '0' cho mỗi thẻ và '1' cho thẻ hiện tại, độ dài của one-hot-vector sẽ bằng số lượng từ trong một từ điển



Build neural network

Ở đây chúng ta xây dựng một mạng nơ-ron bao gồm 1 lớp đầu vào (input layer), 1 lớp đầu ra (output layer), 4 lớp ẩn (hidden layer)

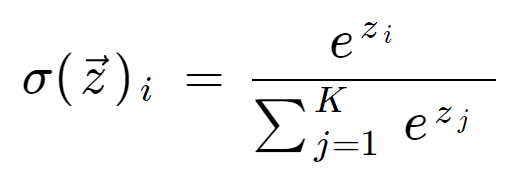


Khởi tạo giá trị bias bằng 0 để phần xử lý có thể đơn giản nhất và dễ hiểu nhất

SOFTMAX: Hàm softmax tính toán xác suất xảy ra của một sự kiện. Hàm softmax sẽ tính khả năng xuất hiện của một class trong tổng số tất cả các class có thể xuất hiện. Sau đó, xác suất này sẽ được sử dụng để xác định class mục tiêu cho các input.

SOFTMAX: Xác suất sẽ luôn nằm trong khoảng (0:1].

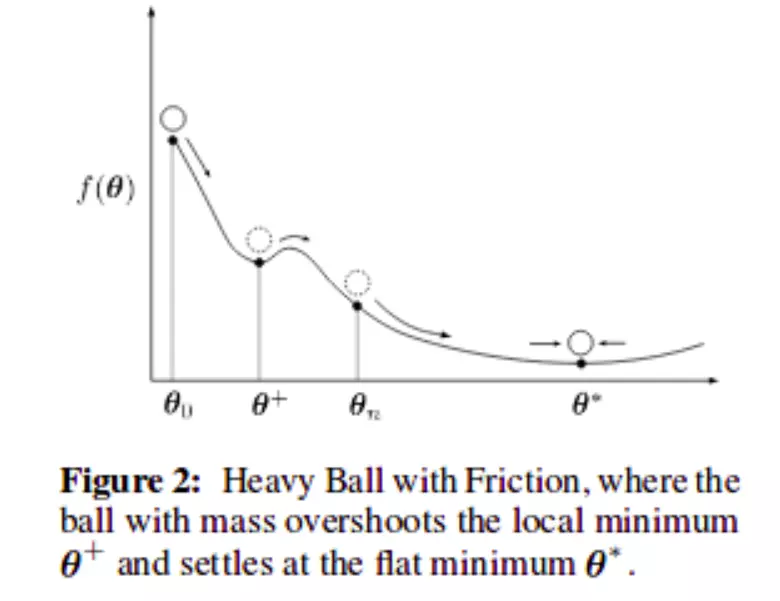
SOFTMAX: Chuẩn hóa softmax là một cách để giảm thiểu ảnh hưởng của những giá trị cực trị hay dữ liệu ngoại lai trong dữ liệu mà không phải chỉnh sửa dữ liệu ban đầu.



Optimizer: Tìm 1 cặp weights và bias phù hợp để tối ưu hóa model. Nhưng vấn đề là "học" như thế nào? Cụ thể là weights và bias được tìm như thế nào! Đâu phải chỉ cần random (weights, bias) 1 số lần hữu hạn và hy vọng ở 1 bước nào đó ta có thể tìm được lời giải. Rõ ràng là không khả thi và lãng phí tài nguyên! Chúng ta phải tìm 1 thuật toán để cải thiện weight và bias theo từng bước, và đó là lý do các thuật toán optimizer ra đời. Ở đây chúng ta sử dụng optimizer là Adam

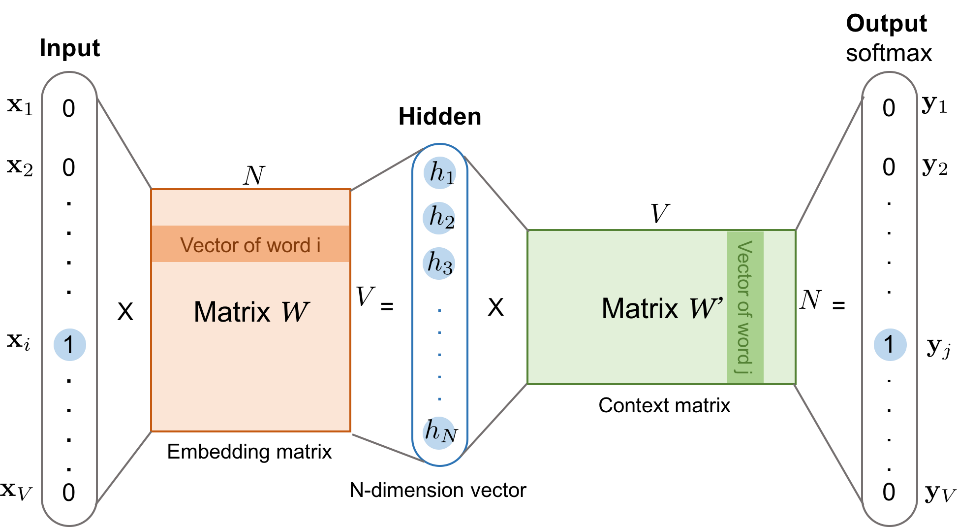
ADAM: là sự kết hợp của Momentum và RMSprop. Adam như 1 quả cầu rất nặng có ma sát, vì vậy nó dễ dàng vượt qua local minimum tới global minimum và khi tới global minimum nó không mất nhiều thời gian dao động qua lại quanh đích vì nó có ma sát nên dễ dừng lại hơn (Mục tiêu là tìm ra giá trị của loss function nhỏ nhất và nhanh nhất. Vì khi loss function càng gần bằng 0 thì định nghĩa càng được khẳng định)

ADAM: Yêu cầu rất ít bộ nhớ, Thích hợp với các bài toán có độ biến thiên không ổn định và dữ liệu traning phân mảnh, yêu cầu ít điều chỉnh



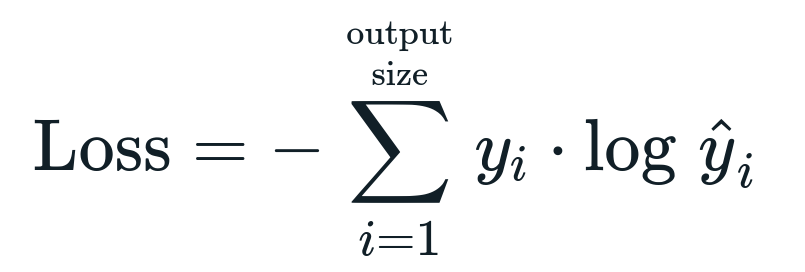
Embedding là một kỹ thuật đưa một vector có số chiều lớn, thường ở dạng thưa, về một vector có số chiều nhỏ, thường ở dạng dày đặc.

Ma trận embedding có thể coi là một ma trận trọng số (weight) trong một mạng neural nhân tạo.



Loss Function: để tối ưu model của mình sao cho tốt nhất, hay cũng dùng để đánh giá độ tốt của model (Giá trị của loss function càng về 0 càng tốt)

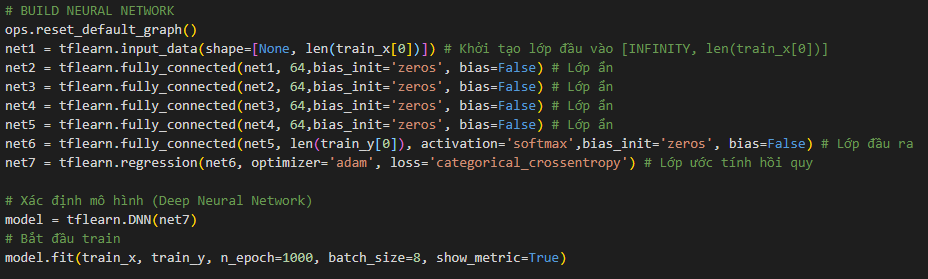
Crossentropy: là độ đo đánh giá độ dài trung bình số lượng bit cần thiết để mã hoá thông điệp với phân bố xác suất là q(x)



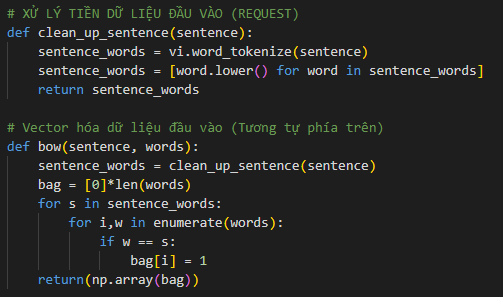
N\_epoch: Lặp qua dataset 1000 lần

Batch\_size: chia tập dữ liệu thành các phần, mỗi phần có kích thước là batch size.

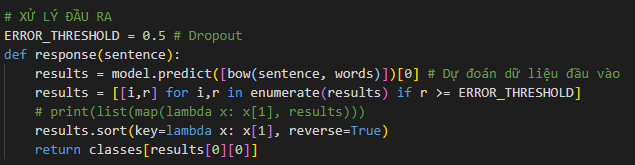
Show\_metric: Show độ chính xác (accuracy)



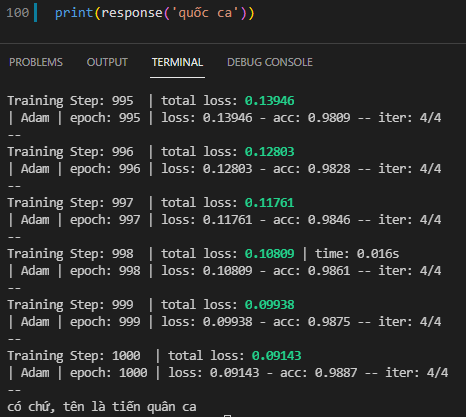
Xử lý tiền dữ liệu đầu vào



Xử lý đầu ra



Kết quả



# III. Các bước thực hiện

## 1. Các bước hoạt động

- Gồm 3 bước:

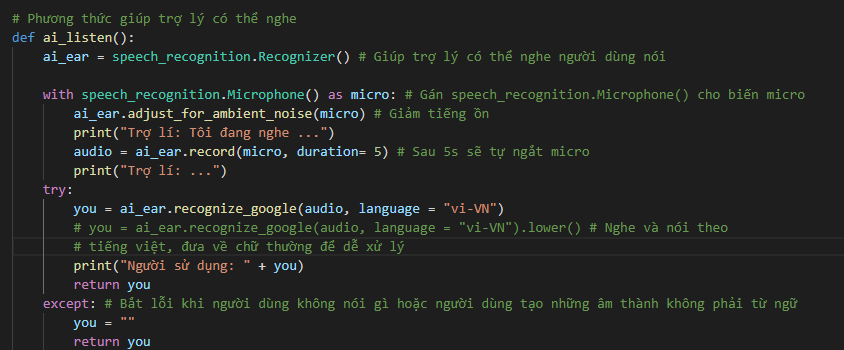
+ Nhận dạng giọng nói, chuyển giọng nói thành văn bản

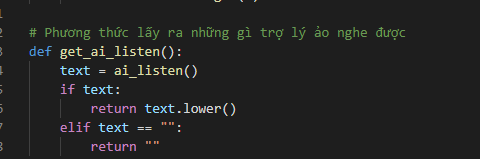
+ Từ văn bản, đưa ra câu trả lời gần đúng nhất

+ Chuyển câu trả lời văn bản thành giọng nói

## 2. Chuyển giọng nói thành văn bản

- Để thực hiện bước này mình sẽ sử dụng thư viện (API) của google là **Speech Recognition.**

- Tiếp theo, để SpeechRecognition xử lý phần Microphone bạn cần cài thêm thư viện pyaudio.

Phương thức lấy ra những gì máy nghe được từ người dùng:

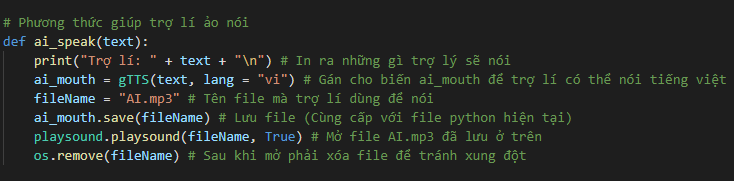
## 3. Xử lý yêu cầu

- Truyền văn bản text đã được chuyển từ giọng nói từ người dùng, đưa vào các phương thức để xử lý yêu cầu và trả về một câu trả lời là một đoạn văn bản.

## 4. Chuyển văn bản thành giọng nói

- Để chuyển văn bản thành giọng nói, ta import thư viện gTTS của google.

- Cài thêm thư viện playsound để phát âm thanh trực tiếp cho python

Ta có phương thức thực hiện như sau:

## 5. Tổng kết

Đây mới chỉ là một trợ lý ảo cơ bản bằng ngôn ngữ python. Vấn đề mà nó gặp phải: thường xuyên không nhận dạng được chính xác từ ngữ người dùng; còn rất đơn sơ, chỉ làm được những vấn đề cơ bản; nó chưa có vẻ thông minh cho lắm; chỉ mới áp dụng được kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) và Machine Learning cơ bản.