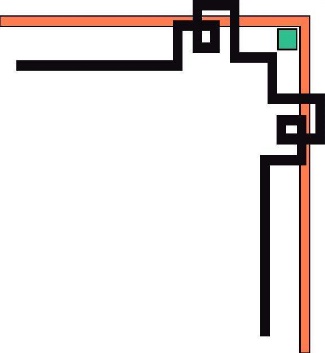
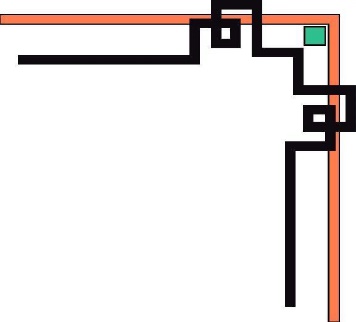
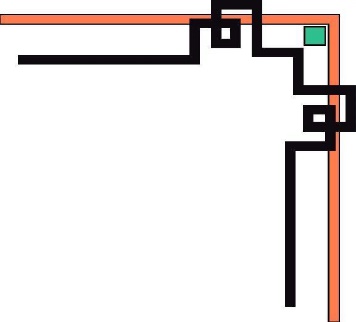
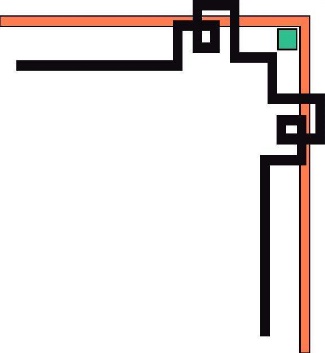
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**



**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN HỌC PHẦN**

TÊN HỌC PHẦN: **HỌC MÁY (Machine Learning: ML = Máy học)**

MÃ SỐ LỚP HP: **MALE431984\_01**

Tên đề tài: **PHÁT TRIỂN HỆ THỐNG HỌC MÁY PHÂN LOẠI PHẢN HỒI CỦA KHÁCH HÀNG DÙNG GIẢI THUẬT MẠNG NƠ-RON NHÂN TẠO**

Họ tên sinh viên: **Nguyễn Trường An**

Mã số sinh viên**: 18110246**

Lớp**: 181103B**

Ngày nộp**: ……./……./202…**

Ký tên**:**

**TP.HCM, ngày …. Tháng … năm 202**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI ĐỒ ÁN HỌC PHẦN**

**Tên HP: HỌC MÁY Mã lớp HP: MALE431984\_01**

Tên đề tài: **PHÁT TRIỂN HỆ THỐNG HỌC MÁY PHÂN LOẠI PHẢN HỒI CỦA KHÁCH HÀNG DÙNG GIẢI THUẬT MẠNG NƠ-RON NHÂN TẠO.**

**Giảng viên giảng dạy: VÕ XUÂN THỂ**

**Sinh viên thực hiện: NGUYỄN TRƯỜNG AN**

**Mã số sinh viên: 18110246**

**Lớp: 181103B**

**Công cụ sử dụng (phiên bản): Python 3.8**

**Ngôn ngữ lập trình sử dụng: Python**

**Nhận xét của giảng viên:**

**………………………………………………………..**

**………………………………………………………..**

**………………………………………………………..**

**Điểm đánh giá:………….(…………………)**

**Ngày……./……../202…..**

**Giảng viên:**

**Ký tên**

**TP.HCM, ngày ……. Tháng … năm 202**

**LỜI CẢM ƠN**

Lời đầu tiên, em xin chân thành cảm ơn thầy Võ Xuân Thể, giảng viên hướng dẫn môn Học máy – người đã mang lại những tiết học bổ ích cho sinh viên. Chính nhờ chỉ dạy tận tình của thầy mà em đã có động lực mạnh mẽ để thực hiện đề tài này.

**DANH MỤC CÁC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
|  | |
| AI | Artificial Intelligence: Trí tuệ nhân tạo |
| ANN | Artificial Neural Network: Mạng nơ-ron nhân tạo |
| ML | Machine Learning: Học máy = Máy học |

**DANH MỤC CÁC THUẬT NGỮ**

|  |  |
| --- | --- |
| Hồi quy tuyến tính | Là một phương pháp phân tích quan hệ giữa biến phụ thuộc Y với một hay nhiều biến độc lập X. |
| Lực | Là bất kỳ ảnh hưởng nào làm một vật thể chịu sự thay đổi, hoặc là ảnh hưởng đến chuyển động, hướng của nó hay cấu trúc hình học của nó |
| Quán tính | Quán tính là tính chất giữ nguyên chuyển động của một vật khi không có lực tác dụng và chỉ thay đổi dần chuyển động nếu có lực tác dụng |

**MỤC LỤC**

[Chương 1: GIỚI THIỆU VỀ ĐỀ TÀI 8](#_Toc60447400)

[Chương 2: CƠ SỞ LÝ THUYẾT VỀ HỌC MÁY 10](#_Toc60447401)

[Chương 3: GIỚI THIỆU VỀ GIẢI THUẬT ML 11](#_Toc60447402)

[Chương 4: GIỚI THIỆU: CÀI ĐẶT VÀ THỰC NGHIỆM HỆ THỐNG ML 15](#_Toc60447403)

[Chương 5: KẾT QUẢ THỰC NGHIỆM HỆ THỐNG ML 18](#_Toc60447404)

[Chương 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI 19](#_Toc60447405)

[6.1. Kết luận 19](#_Toc60447406)

[6.1.1. Những kết quả đạt được 19](#_Toc60447407)

[6.1.2. Hạn chế 19](#_Toc60447408)

[6.2. Hướng phát triển 19](#_Toc60447409)

[6.2.1. Hướng khắc phục các hạn chế 19](#_Toc60447410)

[6.2.2. Hướng mở rộng đề tài 19](#_Toc60447411)

[DANH MỤC TÀI LIỆU THAM KHẢO 20](#_Toc60447412)

[ĐÁNH GIÁ KẾT QUẢ HỌC TẬP HỌC PHẦN 21](#_Toc60447413)

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**PHIẾU GIAO NHIỆM VỤ ĐỀ TÀI ĐỒ ÁN HỌC PHẦN**

|  |  |
| --- | --- |
| **Giảng viên: VÕ XUÂN THỂ** | |
| **Tên học phần**: Học máy **Mã lớp HP:** MALE431984\_01 | |
| **Tên đề tài:** PHÁT TRIỂN HỆ THỐNG HỌC MÁY PHÂN LOẠI PHẢN HỒI CỦA KHÁCH HÀNG DÙNG GIẢI THUẬT MẠNG NƠ-RON NHÂN TẠO | |
| **Sinh viên thực hiện:** Nguyễn Trường An  MSSV 1811246 | |
| **Thời gian thực hiện: 28/09/2020** đến **05/01/2021** | |
| **Yêu cầu của đề tài**  **Lý thuyết:**   1. Căn bản về “học máy”: ML 2. Các giải thuật ML có giám sát (có hướng dẫn) 3. Các giải thuật ML không giám sát (không hướng dẫn) 4. Python và các thư viện ML trên Python 5. Giải thuật Hồi quy tuyến tính và ứng dụng 6. Giải thuật Phân lớp theo láng giềng gần nhất kNNC và ứng dụng 7. Giải thuật Mạng Nơ-ron nhân tạo lan truyền ngược BP\_ANN và ứng dụng 8. Học không giám sát với giải thuật phân cụm (Clustering) k-means và ứng dụng 9. Các giải thuật ML khác   **Thực hành** PHÁT TRIỂN HỆ THỐNG HỌC MÁY PHÂN LOẠI PHẢN HỒI CỦA KHÁCH HÀNG DÙNG GIẢI THUẬT MẠNG NƠ-RON NHÂN TẠO  + Phần chức năng của cá nhân: Thư viện chính được dùng: keras; Tập học, tập thực nghiệm: Phản hồi của khách hàng. Nội dung chính của giải thuật được dùng: ANN | |
| **GIẢNG VIÊN** | **Ngày … tháng … năm 202….**  **SV Thực hiện** |

# 

# GIỚI THIỆU VỀ ĐỀ TÀI

* 1. **Tổng quan về đề tài**
* Đề tài tập trung phát triển hệ thống học máy dùng để phân loại phản hồi của khách hàng là tốt hay xấu
* Giải thuật chính: ANN
* Thư viện: keras, pandas, matplotlib, re, nltk, sklearn
* Tập dữ liệu: 1335 mẫu chia thành 2 phần: tập học và tập thực nghiệm
* Tập học: 1068 mẫu (0.8 lần tập dữ liệu)
* Tập thực nghiệm: 267 mẫu (0.2 lần tập dữ liệu)
* Đặc trưng: tập dữ liệu có 2 thuộc tính: review và liked

1. Thuộc tính review: (dạng text) thể hiện phản hồi của khách hàng về dịch vụ của nhà hàng
2. Thuộc tính liked: (dạng số) thể hiện phản hồi đó là tích cực (1) hay tiêu cực (0)

* 10 mẫu đầu tiên của tập dữ liệu

|  |  |
| --- | --- |
| Wow... Loved this place. | 1 |
| Crust is not good. | 0 |
| Not tasty and the texture was just nasty. | 0 |
| Stopped by during the late May bank holiday off Rick Steve recommendation and loved it. | 1 |
| The selection on the menu was great and so were the prices. | 1 |
| - They never brought a salad we asked for. | 0 |
| Honeslty it didn't taste THAT fresh.) | 0 |
| The potatoes were like rubber and you could tell they had been made up ahead of time being kept under a warmer. | 0 |
| The fries were great too. | 1 |
| A great touch. | 1 |

* 1. **Nội dung chuyên môn chính của đề tài**
* Mô hình túi từ: là mô hình dùng để biểu diễn văn bản dưới dạng một tập hợp (túi) các từ cấu thành nên văn bản mà không quan tâm đến ngữ pháp cũng như thứ tự trước sau của tập các phần tử (từ) trong túi [[1]](#footnote-1).

Để đơn giản hóa mô hình túi từ, chúng ta cần phải làm “sạch” văn bản trước khi đưa vào túi từ bằng cách:

* + - 1. Loại ra khỏi văn bản các kí tự không phải là chữ.
      2. Đưa toàn bộ văn bản về chữ in thường.
      3. Đưa toàn bộ các từ đã được chia theo ngữ pháp về từ nguyên mẫu.
      4. Loại bỏ các từ quá phổ biến, không có ý giá trị phân loại.

Sau khi đã làm sạch văn bản, chúng ta cần có một cách để chuyển văn bản thành véc-tơ để đưa vào đầu vào của ANN

Ở đề tài này, tôi sử dụng phương pháp CountVectorizer để chuyển một đoạn text thành một véc-tơ. Cụ thể sẽ được mô tả ở chương 3.

* ANN: hệ thống mô phỏng hệ thống nơ-ron của con người

Xây dựng hệ thống các nơ-ron để “học” từ tập dữ liệu.

Cấu trúc ANN: chia làm 3 tầng: tầng input, tầng ẩn, tầng output.

Cách học thông qua quá trình điều chỉnh trọng số cho các liên kết giữa các nơ-ron.

* 1. **Mục tiêu của đề tài**

+ Về công nghệ: Áp dụng giải thuật: ANN

+ Về thực tiễn ứng dụng: Có thể sử dụng cho các hệ thống, dịch vụ cần đánh giá phản hồi của khách hàng: các trang thương mại điện tử, các dịch vụ nhà hàng, khách sạn

* 1. **Bố cục của báo cáo**

Chương 1 Giới thiệu tổng quan về đề tài

Chương 2 Cơ sở lý thuyết của học phần: gồm những lý thuyết cơ bản về ML liên quan đến đề tài.

Chương 3 Giới thiệu về Giải thuật ML: ANN

Chương 4 Giới thiệu về cài đặt và thực nghiệm hệ thống ML: Phân loại phản hồi của khách hàng

Chương 5 Kết quản thực nghiệm hệ thống ML: Phân loại phản hồi của khách hàng

Chương 6 Kết luận về kết quản đạt được và những tồn tại, trên cơ cở đó đề xuất các giải pháp khắc phục tồn và hướng mở rộng đề tài.

# CƠ SỞ LÝ THUYẾT VỀ HỌC MÁY

Học máy là một lĩnh vực chuyên sâu của AI tập trung phát triển các hệ thống có khả năng “học”.

Phân loại các giải thuật ML

Dựa trên phương thức học (learning style): 6 loại chia trong 4 nhóm

1. Học có giám sát (Supervised Learning)

1.1. Dùng phương pháp “Phân lớp | loại”: Classification

1.2. Dùng phương pháp “Hồi quy”: Regression

2. Học KHÔNG giám sát (Unsupervised Learning)

2.3. Dùng phương pháp “Phân cụm | nhóm”: Clustering

2.4. Dùng phương pháp “Luật kết hợp”: Association

3. Học BÁN giám sát (Semi-Supervised Learning)

4. Học Tăng cường || củng cố (Reinforcement Learning)

# GIỚI THIỆU VỀ GIẢI THUẬT ML

* Mô hình túi từ:

Ví dụ tôi có một câu như sau: “The cat is eating the fish”

Bước 1: đánh số cho từng từ trong câu (các từ giống nhau có cùng số)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| THE | CAT | IS | EATING | THE | FISH |
| 0 | 1 | 2 | 3 | 0 | 4 |

Bước 2: Đếm số lần xuất hiện của từ được đánh số trong câu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 2 | 1 | 1 | 1 | 1 |

Bước 3: véc-tơ nhận được sau khi đếm chính là véc-tơ sau khi thực hiện CountVectorizer

Kết quả: “The cat is eating the fish” đã được véc-tơ hóa thành véc-tơ [2,1,1,1,1] và véc-tơ này có thể đưa vào hệ thống học máy.

* ANN

Cấu trúc ANN(dạng full connection):

+ Số lượng tín hiệu Đầu vào: bằng với số lượng các từ có trong mô hình túi từ được định nghĩa ở trên

+ Số lượng tín hiệu Đầu ra: 1 – là tích cực (1) hay tiêu cực (0)

+ Số lượng tầng: Một tầng đầu vào, một tầng đầu ra, 4 tầng ẩn.

+ Số lượng nơ-ron mỗi tầng: Mỗi tầng ẩn có số lượng nơ-ron lần lượt theo thứ tự là: 300, 100, 20, 10

+ Số lượng các trọng số: mỗi nơ-ron có liên kết với tất cả nơ-ron ở tầng kế tiếp. Mỗi liên kết có một trọng số

+ Cách thức các nơ-ron liên kết: Mỗi nơ-ron có liên kết đến tất cả các nơ ron ở tầng kế tiếp

Cách học của ANN: Điều chỉnh trọng số

Giải thuật điều chỉnh trọng số: Adam

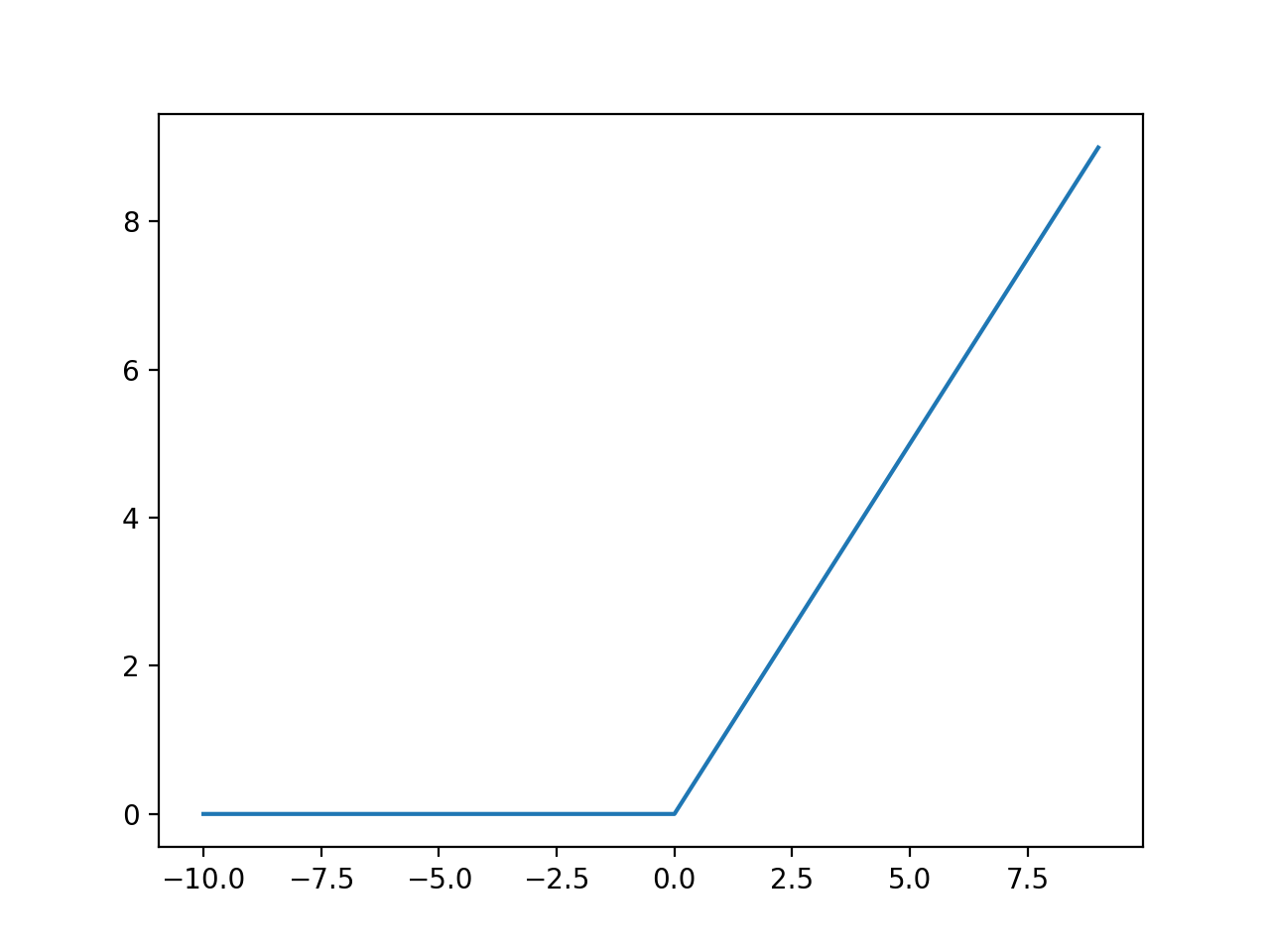
Activation function: ReLU, sigmoid (được dùng ở tầng output).

Hàm đánh giá lỗi: binary\_crossentropy

Epochs: 100

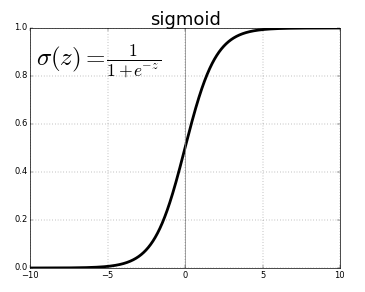
Batch\_size: 10

* ReLU Activation Function



Công thức hàm ReLU

* Sigmoid Activation Function



Công thức hàm sigmoid

* Giải thuật điều chỉnh trọng số Adam

Công thức tổng quát

Với

Với

: (đọc là Ép-si-lon) giá trị cực nhỏ, để tránh trường hợp chia cho 0, thường là 10-8

: (đọc là theta) trọng số được cập nhật lần thứ t

: (đọc là eta) tốc độ học

: (gradient) đạo hàm theo biến theta thứ t

: chỉ số quán tính thứ 1 ()

: chỉ số quán tính thứ 2 ()

: hệ số phụ thuộc (0.9)

: hệ số phụ thuộc (0.999)

Ý tưởng giải thuật: Được hình thành để khác phục hạn chế của 2 thuật toán tối ưu Momentum và Adapvite Learning Rate

* Momentum

Sau mỗi lần học, trọng số sẽ được điều chỉnh để đạt được output tối ưu.

Ý tưởng: việc tìm điểm tối ưu như quả cầu đang lau dốc. Với momentum, chúng ta sẽ áp dụng lực quán tính với quả cầu này.

Công thức chung:

Với

Ta được

Với

độ dốc của điểm trước đó (đạo hàm của hàm mất mát với biến là )

: (đọc là gamma) là hằng số quán tính (thường chọn là 0.9)

: xem như lực quán tính của quả cầu.

Như vậy với Momentum, chúng ta nâng cao được khả năng hội tụ của bài toán. Nhưng vẫn còn hạn chế, đó là: khi gần đạt được giá trị tối ưu, thuật toán vẫn khá khó khăn để đạt được vị trí tối ưu này (vì còn lực quán tính ).

* Adapvite Learning Rate

Tiếp cận với trường phái khác của việc xử lí bài toán tối ưu, lần này chúng ta sẽ cập nhật lại giá trị để việc hội tụ trở nên nhanh chóng hơn, tôi xin lấy điển hình thuật toán AdaGrad, RMSprop

Với ý tưởng trên, AdaGrad có công thức chung như sau:

Công thức có điểm khác biệc so với thuật toán tối ưu GD, đó chính là được chia cho . (với là hằng số bé để tránh trường hợp chia với 0)

Mỗi lần cập nhật trọng số, sẽ càng nhỏ. (vì sẽ càng lớn với t càng lớn).

Vì thế, sẽ có trường hợp tiến nhanh về 0. Khi đó tốc độ học bị đóng băng.

Để tránh việc này. Thuật toán RMSprop đã hình thành, cụ thể như sau:

sẽ được thay thế bằng

Ví dụ với t = 3 ta tính được như sau,

Còn được xác định như sau:

Dễ nhận thấy, < vì thế khả năng bị đóng băng tốc độ học được hạn chế

# GIỚI THIỆU: CÀI ĐẶT VÀ THỰC NGHIỆM HỆ THỐNG ML

Sử dụng một số thư viện cần thiết

#%% Importing the libraries  
import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd  
import keras  
from keras import Sequential

Đọc dữ liệu từ file tsv

dataset = read\_dataset('Restaurant\_Reviews.tsv')

Nhập các từ không có nhiều ý nghĩa phân loại (file đã được loại bỏ các từ mang ý nghĩa phủ định)

#%% mở file chứa stopwords  
stopwords = []  
file = open('stopwords.txt', "r")  
try:  
 content = file.read()  
 stopwords\_list = content.split(",")  
 for stopword in stopwords\_list:  
 s = stopword.replace('"','')  
 s = s.strip()  
 stopwords.append(s)  
finally:  
 file.close()

Xử lí text

corpus = [] #chứa các reviews đã qua các bước lọc  
for i in range(0, dataset.shape[0]):  
 review = re.sub('[^a**-**zA**-**Z]', ' ', dataset['Review'][i]) #loại bỏ các phần không phải  
 #là chữ cái, thay thế bằng dấu spaces  
 #^: not  
 #a-zA-Z: a to z nor A to Z  
 #' ': space  
 review = review.lower() # all to lowercase  
 review = review.split() # split to words  
 ps = PorterStemmer() # ran => run,....  
 review = [ps.stem(word) for word in review if not word in set(stopwords)] # chuyển về nguyên  
 # mẫu các từ không có trong stopwords  
 review = ' '.join(review) # nối lại các từ thành câu  
 corpus.append(review)

Tạo mô hình túi từ

#%% Creating the Bag of Words model  
from sklearn.feature\_extraction.text import CountVectorizer  
cv = CountVectorizer(max\_features = 2000) # chọn ra 2000 từ  
X = cv.fit\_transform(corpus).toarray()   
y = dataset.iloc[:, -1].values

Chia túi từ thành 2 tập để học và để thực nghiệm

#%% Splitting the dataset into the Training set and Test set  
from sklearn.model\_selection import train\_test\_split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.20, random\_state = 1)

Xây dựng Mạng nơ-ron nhân tạo

#%% ANN model  
from keras import layers  
model = Sequential()   
input\_dim = X\_train.shape[1] # input đầu vào bằng với số từ của tập dữ liệu  
model = Sequential()  
model.add(layers.Dense(300, input\_dim=input\_dim, activation='relu'))  
model.add(layers.Dense(100, activation='relu'))  
model.add(layers.Dense(20, activation='relu'))  
model.add(layers.Dense(10, activation='relu'))  
model.add(layers.Dense(1, activation='sigmoid')) # đầu ra là giá trị từ 0 đến 1  
  
model.compile(loss='binary\_crossentropy',   
 optimizer='adam',   
 metrics=['accuracy'])

Huấn luyện Mạng nơ-ron từ tập học

history = model.fit(X\_train, y\_train,  
 epochs=100,  
 verbose=1,  
 validation\_data=(X\_test, y\_test),  
 batch\_size=10)

Đánh giá quá trình học

loss1, accuracy1 = model.evaluate(X\_train, y\_train, verbose=1)  
loss2, accuracy2 = model.evaluate(X\_test, y\_test, verbose=1)  
print("Training Accuracy: {:.4f}".format(accuracy1))  
print("Testing Accuracy: {:.4f}".format(accuracy2))

Xuất ra dạng đồ thị

acc = history.history['accuracy']  
val\_acc = history.history['val\_accuracy']  
loss = history.history['loss']  
val\_loss = history.history['val\_loss']  
epochs = range(1,len(acc)+1)  
  
plt.plot(epochs, acc,'bo', label='Training accuracy')  
  
plt.plot(epochs, val\_acc,'b', label='Validation accuracy')  
plt.ylim([0,1])  
plt.title('')  
plt.legend()  
plt.show()

Hàm để chuyển 1 review thành dạng véc-tơ (dùng để dự đoán)

def preprocess\_review\_input(review):  
 new\_review = review  
 new\_review = re.sub('[^a**-**zA**-**Z]', ' ', new\_review)  
 new\_review = new\_review.lower()  
 new\_review = new\_review.split()  
 ps = PorterStemmer()  
 new\_review = [ps.stem(word) for word in new\_review if not word in set(stopwords)]  
 new\_review = ' '.join(new\_review)  
 new\_corpus = [new\_review]  
 new\_X\_test = cv.transform(new\_corpus).toarray()  
 return new\_X\_test

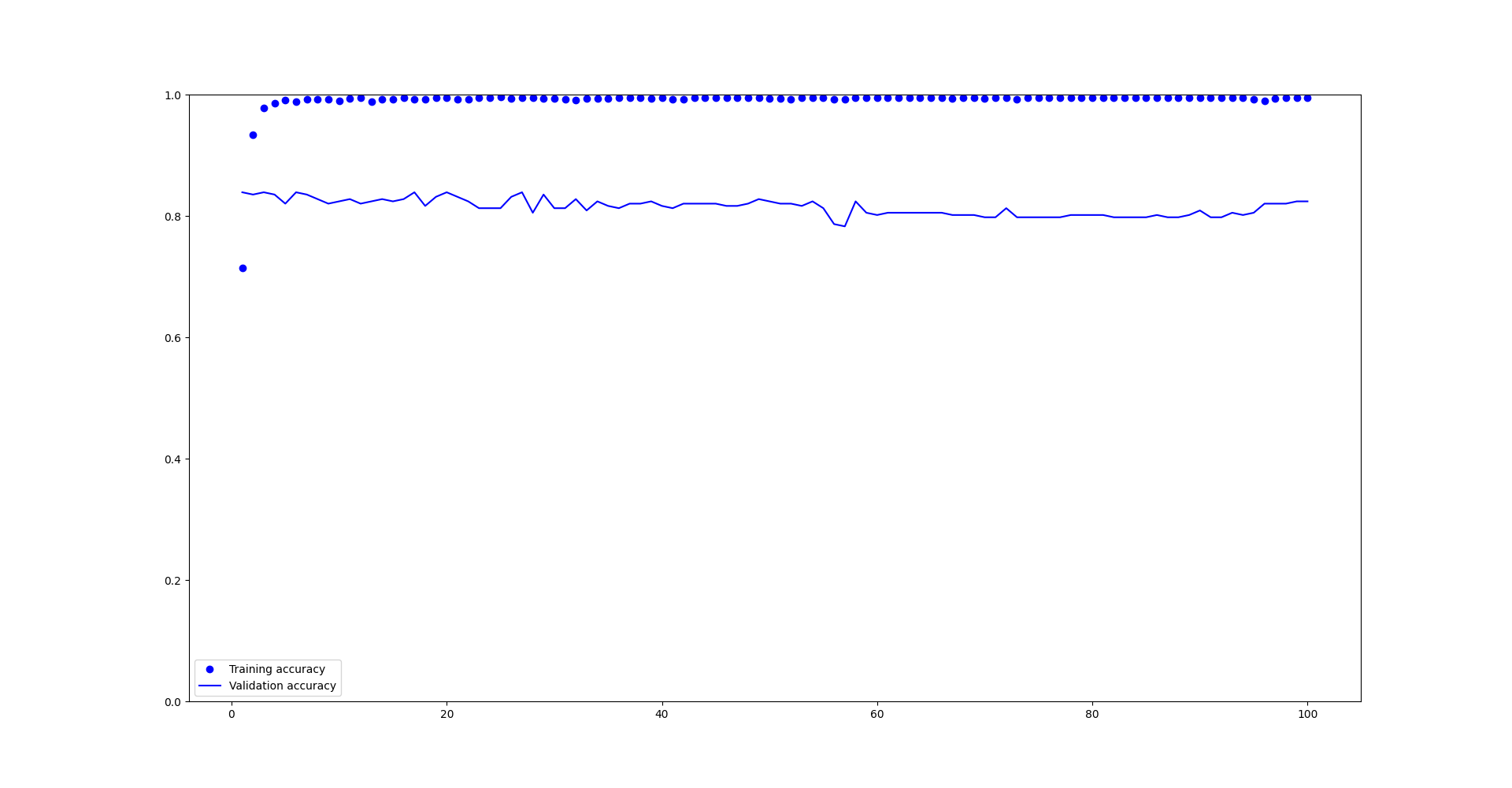
Hàm dùng để đánh giá 1 phản hồi

def review\_input(review):  
 y\_pred=model.predict(preprocess\_review\_input(review))  
 print(y\_pred)  
 return y\_pred[0][0]

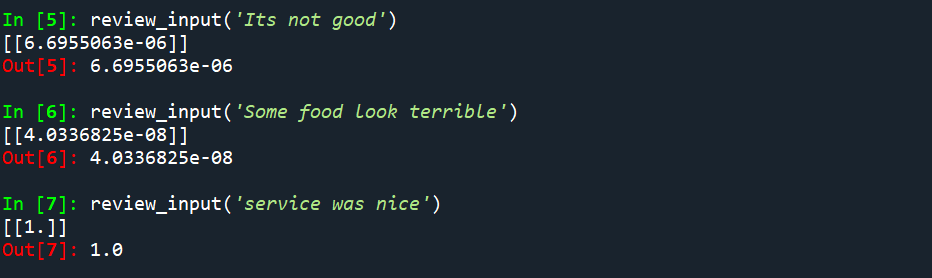
# KẾT QUẢ THỰC NGHIỆM HỆ THỐNG ML

Kết quả





Kết quả dự đoán 1 câu phản hồi



# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

## **Kết luận**

### Những kết quả đạt được

* Phân loại được các phản hồi theo hướng tích cực hay tiêu cực (có độ chính xác hơn 80%)

### Hạn chế

Nêu những vấn đề còn tồn tại.

* Ở các trường hợp phản hồi theo hướng trung lập (không khen, không chê hoặc vừa có khen vừa có chê), hệ thống vẫn chưa phân loại một cách chính xác.

## **Hướng phát triển**

### Hướng khắc phục các hạn chế

* Mở rộng tập học mang tính bao quát cao hơn

### Hướng mở rộng đề tài

* Đánh giá phản hồi của khách hàng dựa trên thang điểm 10
* Đánh giá phản hồi của khách hàng trên một lĩnh vực khác (ngoài nhà hàng)

# DANH MỤC TÀI LIỆU THAM KHẢO

1. Võ Xuân Thể (2020), *Bài giảng học phần Học máy*, TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM (lưu hành nội bộ).

# ĐÁNH GIÁ KẾT QUẢ HỌC TẬP HỌC PHẦN

ĐỒ ÁN HP THAY CHO BÀI THI KẾT THÚC HỌC PHẦN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Hình thức KT** | **Nội dung** | **Chuẩn đầu ra đánh giá** | **Trình độ năng lực** | **Phương pháp đánh giá** | **Công cụ đánh giá** | **Tỉ lệ (%)** |
| **BÀI TẬP LẬP TRÌNH** | | | | | |  |
| BL#1 | Bài tập lập trình theo từng Buổi thực hành = tính bình quân | G1.1 | 3 | Bài tập | Bài tập | 40 |
| G2.1 | 3 |
| G3.1  G3.2 | 4  3 |
| G4.1 | 5 |
| **ĐỒ ÁN HỌC PHẦN (Project)** | | | | | |  |
| ĐA#1 | Phát triển một hệ thống học máy đơn giản:  + Đề tài chung theo nhóm: Giải thuật ML  + Xây dựng và thực nghiệm giải thuật theo bài toán cá nhân  => báo cáo vào tuần cuối. | G1.1 | 3 | Báo cáo, Demo và thuyết trình | Rubric | 40 |
| G2.1 | 3 |
| G3.1 | 4 |
| G3.2  G4.1 | 3  5 |
| G2.1 | 4 |
| G3.1 | 4 |
| G3.2  G4.1 | 3  5 |
| **BÀI KIỂM TRA LÝ THUYẾT** | | | | | |  |
| BT#1 | Bài kiểm tra lý thuyết: Hiểu các giải thuật  học máy phổ biến?**CĐR 2.2** | G1.1 | 3 | Kiểm tra | Câu hỏi | 20 |
| G2.1 | 3 |
| G3.2 | 3 |

1. Đây là khái niệm của tác giả bài báo cáo này. [↑](#footnote-ref-1)