

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - CƠ - TIN HỌC



BÁO CÁO CUỐI KÌ

**Tên đề tài: PHÁT HIỆN TIN THẬT TIN GIẢ SỬ DỤNG
MẠNG NƠ-RON TÍCH CHẬP (CNN)**

Nhóm thực hiện: Nhóm 3

PHƯƠNG PHÁP NGHIÊN CỨU KHOA HỌC

Hà Nội - 12/2024

Báo Cáo Cuối Kỳ

Phương pháp nghiên cứu khoa học

Giảng viên hướng dẫn: TS. Lê Huy Hoàng

Sinh viên thực hiện: Nguyễn Thành Trung - 22001672

Nguyễn Thị Ánh - 22000070

Nguyễn Tiến Đạt - 22000081

Ngày 31 tháng 12 năm 2024

Tóm tắt nội dung

Nội dung bao quát của dự án là ứng dụng mạng nơ-ron tích chập (CNN) trong phát hiện và phân loại tin tức thật, giả – một thách thức lớn trong kỷ nguyên thông tin số. Với khả năng tự động trích xuất các đặc trưng từ dữ liệu, CNN được chứng minh là hiệu quả trong việc phân tích các văn bản tin tức, giúp nhận diện và phân biệt các mẫu tin giả với độ chính xác cao. Phần đầu tiên giới thiệu tổng quan về nền tảng lý thuyết và cơ sở của CNN, bao gồm việc xử lý dữ liệu chuỗi và đánh giá hiệu suất mô hình. Phạm vi nghiên cứu tập trung vào các giai đoạn từ xây dựng, đào tạo mô hình, đến các phương pháp xử lý dữ liệu đầu vào như dọn dẹp và tiền xử lý ngữ liệu với nhúng từ. Dự án cũng mô tả chi tiết các bước thực nghiệm, bao gồm chuẩn bị và xử lý dữ liệu, khởi tạo và huấn luyện mô hình CNN.

Kết quả nghiên cứu cho thấy mô hình CNN đạt được độ chính xác cao trong việc phân loại tin thật, giả, chứng minh tính hiệu quả của phương pháp học sâu trong xử lý ngôn ngữ tự nhiên. Qua đó, báo cáo nhấn mạnh tiềm năng ứng dụng rộng rãi của CNN trong việc giảm thiểu tác động tiêu cực của tin giả đối với xã hội. Kết luận khẳng định rằng việc sử dụng CNN không chỉ cải thiện khả năng nhận diện tin giả mà còn mở ra hướng nghiên cứu mới trong lĩnh vực trí tuệ nhân tạo ứng dụng.

Mục lục

| | | |
|----------|---|-----------|
| 1 | Giới Thiệu và cơ sở mô hình CNN | 2 |
| 1.1 | Đặt vấn đề | 2 |
| 1.2 | Phạm vi nghiên cứu | 3 |
| 1.2.1 | Tập Trung vào CNN | 3 |
| 1.2.2 | Xây Dựng và Đào Tạo Mô Hình | 3 |
| 1.2.3 | Xử Lý Dữ Liệu Chuỗi | 3 |
| 1.2.4 | Đánh Giá Hiệu Suất | 3 |
| 1.3 | Đối tượng nghiên cứu | 4 |
| 1.4 | Mục tiêu nghiên cứu | 4 |
| 1.5 | Cách thức tiến hành | 4 |
| 1.6 | Mạng thần kinh tích chập (Convolutional neural network - CNN) | 6 |
| 1.6.1 | Giới thiệu | 6 |
| 1.6.2 | Nguyên lý hoạt động của CNN | 7 |
| 2 | Mô hình thực nghiệm | 10 |
| 2.1 | Chuẩn bị dữ liệu đầu vào | 10 |
| 2.2 | Dọn dẹp dữ liệu | 11 |
| 2.3 | Tiền xử lý dữ liệu và nhúng từ | 11 |
| 2.4 | Khởi tạo mô hình CNN | 12 |
| 3 | Kết quả mô hình thực nghiệm | 16 |
| 4 | Đánh giá và Thảo Luận | 18 |

Chương 1

Giới Thiệu và cơ sở mô hình CNN

1.1 Đặt vấn đề

Dự án này tập trung vào việc phát triển một mô hình học máy nhằm phân loại và phát hiện tin tức giả, dựa trên hai nghiên cứu khoa học đã công bố: **“Detection of fake news using deep learning CNN-RNN based methods”**[1] của *I. Kadek Sastrawan, I.P.A. Bayupati, Dewa Made Sri Arsa* và **“Fake news detection: A hybrid CNN-RNN based deep learning approach”**[2] của *Jamal Abdul Nasir, Osama Subhani Khan, Iraklis Varlamis*. Sự lan rộng của thông tin sai lệch, thường gọi là tin giả, đang trở thành một vấn đề ngày càng nghiêm trọng trong thời đại kỹ thuật số hiện nay. Với tốc độ lan truyền nhanh chóng thông qua internet và các nền tảng mạng xã hội, môi trường này tạo điều kiện cho tin tức giả phổ biến và ảnh hưởng sâu rộng đến nhận thức công chúng. Bên cạnh đó, sự phát triển của công nghệ, bao gồm trí tuệ nhân tạo và các phương tiện kỹ thuật số, đã làm cho tin giả ngày càng trở nên phức tạp và khó phát hiện hơn. Điều này đặt ra thách thức lớn cho cả người tiêu dùng thông tin và các tổ chức truyền thông trong nỗ lực kiểm chứng và xử lý tin tức.

Vì vậy, dự án đặt trọng tâm vào việc phân tích, tìm hiểu các nghiên cứu khoa học nói trên để xây dựng nền tảng kỹ thuật cho mô hình có khả năng nhận diện và phân biệt tin tức thật và giả, từ đó góp phần giải quyết vấn đề tin giả trong bối cảnh truyền thông hiện đại.

1.2 Phạm vi nghiên cứu

Phạm vi nghiên cứu của dự án này được xác định với mục tiêu giải quyết vấn đề phân biệt tin tức thật và giả trong các bài báo. Nghiên cứu chủ yếu tập trung vào việc phát triển một hệ thống phân loại tin tức giả dựa trên mô hình mạng nơ-ron tích chập (CNN - Convolutional Neural Network). Dưới đây là một phác thảo chi tiết về các khía cạnh chính trong phạm vi nghiên cứu này.

1.2.1 Tập Trung vào CNN

Mô hình CNN được chọn làm trọng tâm của nghiên cứu để khai thác khả năng học đặc trưng không gian của nó. Các lớp tích chập và pooling của CNN có thể hiệu quả trong việc nhận diện mẫu và cấu trúc trong văn bản.

1.2.2 Xây Dựng và Đào Tạo Mô Hình

Nghiên cứu bao gồm quá trình xây dựng và đào tạo một mô hình CNN trên tập dữ liệu ISOT (Integrated Student Outcomes Toolkit). Việc này nhằm mục đích hiểu rõ khả năng áp dụng của mô hình trên một tập dữ liệu thực tế và đa dạng.

1.2.3 Xử Lý Dữ Liệu Chuỗi

Với đặc tính của tin tức, mô hình phải xử lý dữ liệu chuỗi từ ngôn ngữ tự nhiên. Nghiên cứu sẽ tập trung vào cách mô hình học và áp dụng các đặc trưng ngôn ngữ để phân biệt giữa tin tức thật và giả.

1.2.4 Đánh Giá Hiệu Suất

Hiệu suất của mô hình sẽ được đánh giá dựa trên các thước đo như Accuracy, Precision, Recall, và F1-score. Các kết quả này sẽ cung cấp thông tin về khả năng của mô hình trong việc phân loại tin tức và ứng dụng thực tế của nó.

Phạm vi nghiên cứu đề cập đến những khía cạnh quan trọng của việc phát triển một hệ thống phân loại tin tức giả, từ lựa chọn mô hình đến áp dụng thực tế và đa dạng của dữ liệu. Điều này giúp định rõ hướng đi và ý nghĩa của nghiên cứu trong ngữ cảnh rộng lớn của vấn đề tin tức giả trong thế giới số ngày nay.

1.3 Đối tượng nghiên cứu

Đối tượng nghiên cứu của dự án là mô hình CNN được sử dụng để phát hiện và phân loại các bài báo thành tin thật hoặc giả. Nghiên cứu này tập trung vào việc đánh giá tính độc lập và mức độ đáng tin cậy của các nguồn thông tin trong truyền thông, đặc biệt chú trọng đến khả năng phát hiện tin sai lệch và tin giả (fake news detection) trong các ngữ cảnh khác nhau.

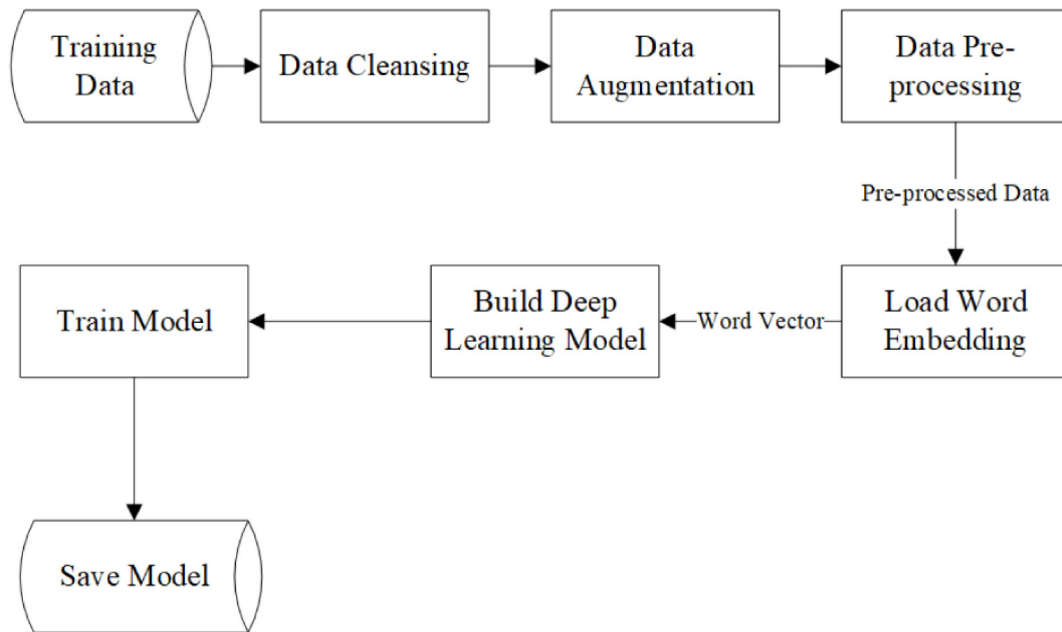
1.4 Mục tiêu nghiên cứu

Mục tiêu chính của nghiên cứu là xây dựng một mô hình phân loại tự động có khả năng xác định các bài báo giả mạo hoặc thiếu độ tin cậy, sử dụng công nghệ CNN mới. Bằng cách này, nghiên cứu hướng đến việc phát triển một công cụ có hiệu quả trong việc giảm thiểu sự lan truyền của tin sai lệch, góp phần nâng cao nhận thức và niềm tin của người đọc đối với các nguồn thông tin.

Ngoài ra, nghiên cứu này còn nhắm đến việc tạo ra một khung lý thuyết tổng quát cho khả năng ứng dụng CNN trong xử lý văn bản và dữ liệu ngôn ngữ tự nhiên. Mục tiêu này nhằm đánh giá tính hiệu quả của CNN trong phân loại tin tức ở nhiều ngữ cảnh và loại dữ liệu khác nhau, giúp mở rộng tiềm năng ứng dụng của mô hình trong các bài toán phân loại tin tức.

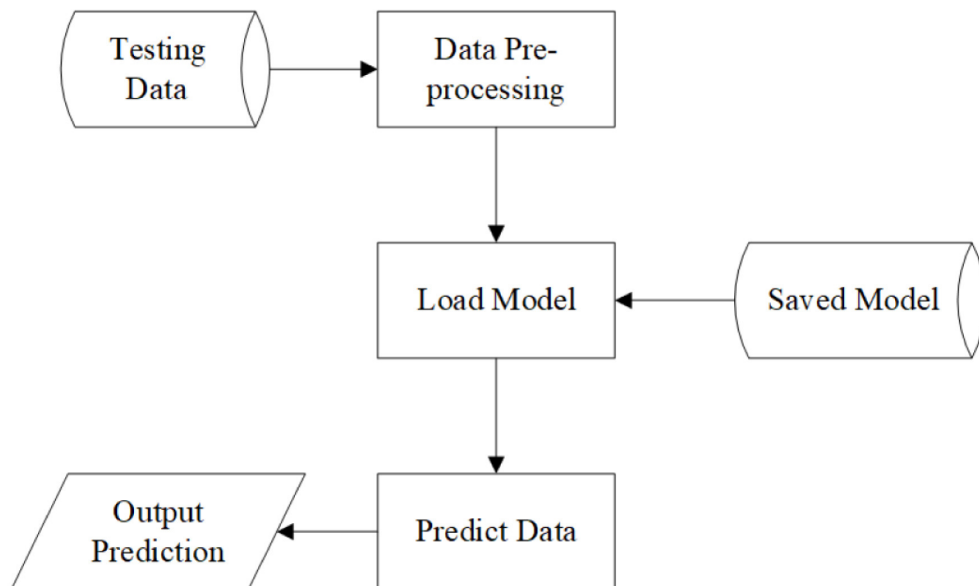
1.5 Cách thức tiến hành

Dự án này trải qua một số giai đoạn, như được mô tả trong Hình 1.1 và 1.2. Nó bắt đầu bằng việc lấy dữ liệu huấn luyện từ cơ sở dữ liệu, sau đó là làm sạch dữ liệu thông qua việc loại bỏ các từ dừng và dấu chấm câu. Bước tiếp theo liên quan đến việc xử lý trước dữ liệu để chuyển đổi dữ liệu văn bản sang dạng số. Dữ liệu số sau đó được chuyển qua lớp nhúng để chuyển đổi nó thành biểu diễn vectơ. Những dữ liệu được vectơ hóa này được đưa vào các mô hình được xây dựng để đào tạo và sau đó, các mô hình này được đánh giá bằng cách sử dụng tập dữ liệu thử nghiệm.



Hình 1.1: Xử lý và huấn luyện (1)

Cùng với quá trình xử lý dữ liệu và huấn luyện mô hình, chúng ta tiếp tục thực hiện bước kiểm thử dữ liệu và lấy dữ liệu đã lưu ở quy trình (1) để tải mô hình và chạy thử dự đoán.



Hình 1.2: Kiểm thử và chạy dữ liệu

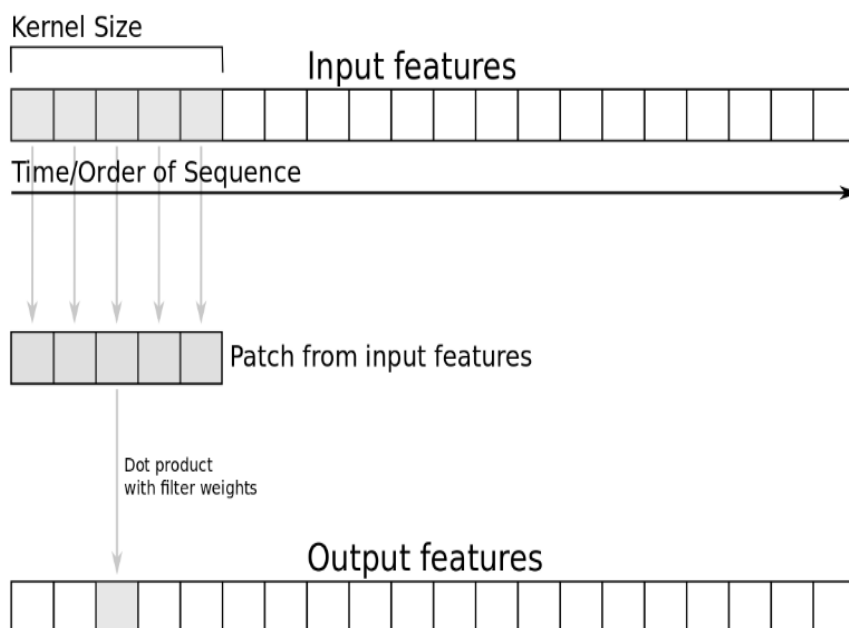
*Nguồn: Tài liệu tham khảo số [1]

1.6 Mạng thần kinh tích chập (Convolutional neural network - CNN)

1.6.1 Giới thiệu

Mạng nơ-ron tích chập (CNN) bao gồm phép nhân ma trận để tạo ra các đầu ra phục vụ cho quá trình huấn luyện tiếp theo. Phương pháp này được gọi là tích chập, và đó là lý do loại mạng này được gọi là mạng nơ-ron tích chập. Trong xử lý ngôn ngữ tự nhiên (NLP), các từ trong câu hoặc bài báo được biểu diễn dưới dạng vectơ từ, sau đó được dùng để huấn luyện CNN. Quá trình huấn luyện được thực hiện bằng cách xác định kích thước hạt nhân và số lượng bộ lọc. CNN có thể đa chiều.

Trong phân loại văn bản hoặc NLP, thường sử dụng CNN một chiều (Conv1D). Conv1D làm việc với các mảng một chiều đại diện cho vectơ từ. Trong CNN, một bộ lọc có kích thước cố định sẽ duyệt qua dữ liệu huấn luyện, tại mỗi bước sẽ nhân dữ liệu đầu vào với trọng số của bộ lọc và tạo ra đầu ra được lưu vào một mảng đầu ra, gọi là bản đồ đặc trưng hoặc bộ lọc đầu ra. Cách này giúp phát hiện đặc trưng từ dữ liệu đầu vào.



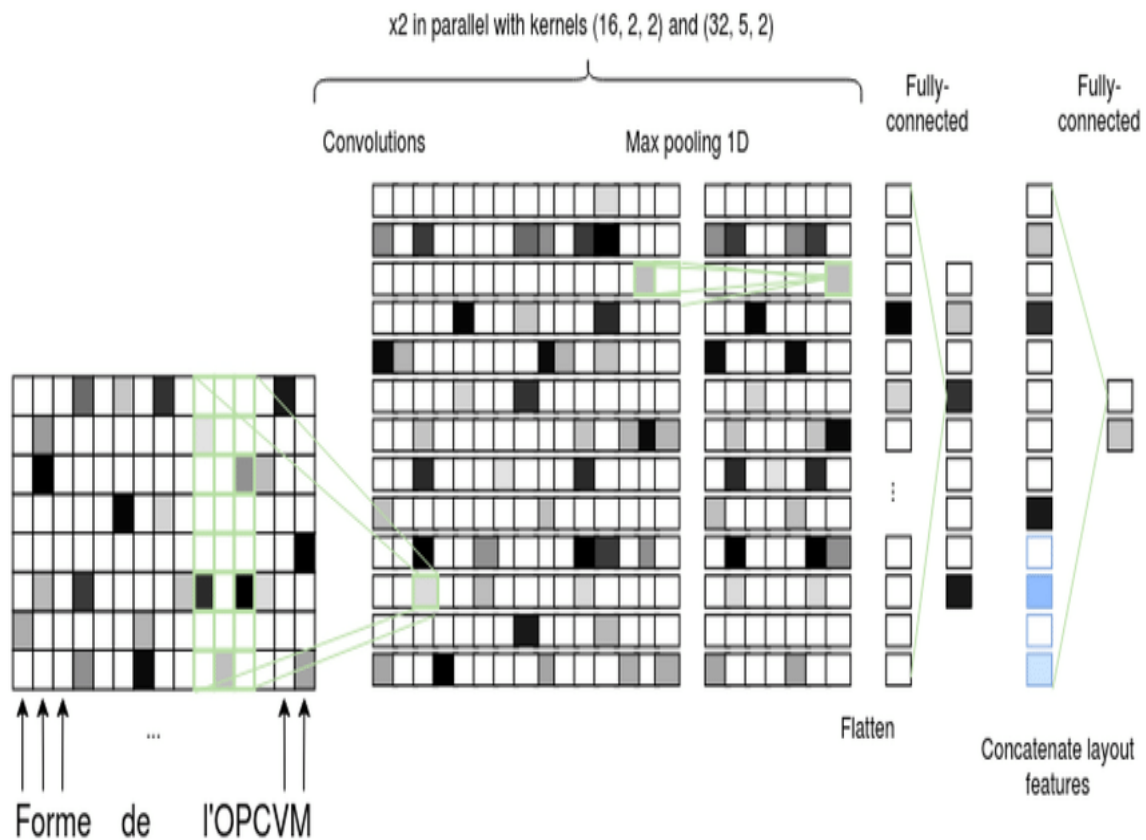
Hình 1.3: Mô phỏng mạng thần kinh tích chập

*Nguồn: <https://realpython.com/python-keras-text-classification/>

1.6.2 Nguyên lý hoạt động của CNN

Cách thức hoạt động:

Mạng CNN là một tập hợp các lớp tích chập chồng lên nhau và sử dụng các hàm kích hoạt phi tuyến như ReLU và tanh để kích hoạt các trọng số trong các nơ-ron. Mỗi lớp, sau khi qua các hàm kích hoạt, sẽ tạo ra các thông tin trừu tượng hơn cho lớp tiếp theo. Trong mô hình CNN, các lớp được kết nối với nhau thông qua cơ chế tích chập, với lớp tiếp theo là kết quả của phép tính tích chập từ lớp trước đó, giúp tạo ra các kết nối cục bộ. Nhờ vậy, mỗi nơ-ron ở lớp kế tiếp được sinh ra từ kết quả của filter áp đặt lên một vùng cục bộ của nơ-ron lớp trước. Mỗi lớp sử dụng các filter khác nhau, thông thường có hàng trăm đến hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Trong quá trình huấn luyện, mạng CNN tự động học các giá trị qua các lớp filter này.



Hình 1.4: Mô hình CNN

*Nguồn: Proposed CharCNN encoder model

Mô hình CNN sẽ được sử dụng trong đề tài Fake new Detection gồm các lớp:

1. Lớp Embedding:

- Mục đích: Biểu diễn từng từ trong câu thành một vector số thực.

- Sử dụng word embeddings (ví dụ: Word2Vec, GloVe) để ánh xạ từng từ về một không gian vector nhiều chiều, giữ lại thông tin ngữ nghĩa và ngữ cảnh của từ.

2. Lớp Convolutional:

- Mục đích: Nhận diện các đặc trưng cấp thấp và cấp cao trong văn bản.
- Áp dụng một số lớp Conv1D để quét qua các chuỗi từ với các bộ lọc (kernels) để trích xuất các đặc trưng quan trọng. Các bộ lọc này có thể nhận biết các mô hình ngôn ngữ hoặc mô hình ngữ cảnh.

3. Lớp Max Pooling:

- Mục đích: Giảm kích thước của đầu ra và giữ lại đặc trưng quan trọng.
- Với mỗi đoạn văn sau khi lớp Convolutional, áp dụng lớp Max Pooling để chọn giá trị lớn nhất từ mỗi cửa sổ, giảm kích thước của đầu ra và tăng tính trừu tượng.

4. Lớp Flatten:

- Mục đích: Chuyển đổi đầu ra từ các ma trận 2D sang vector 1D để chuẩn bị cho lớp Fully Connected.
- Đặt tất cả các giá trị từ ma trận thành một vector để nó có thể được sử dụng làm đầu vào cho các lớp Fully Connected.

5. Lớp Fully Connected (Dense):

- Mục đích: Học các mối quan hệ phức tạp giữa các đặc trưng.
- Các lớp Dense chứa các nơ-ron được kết nối đầy đủ với nơi đầu vào từ các lớp trước. Các lớp này học các biểu diễn phức tạp và mối quan hệ giữa các đặc trưng để có thể phân loại giả mạo hay không giả mạo.

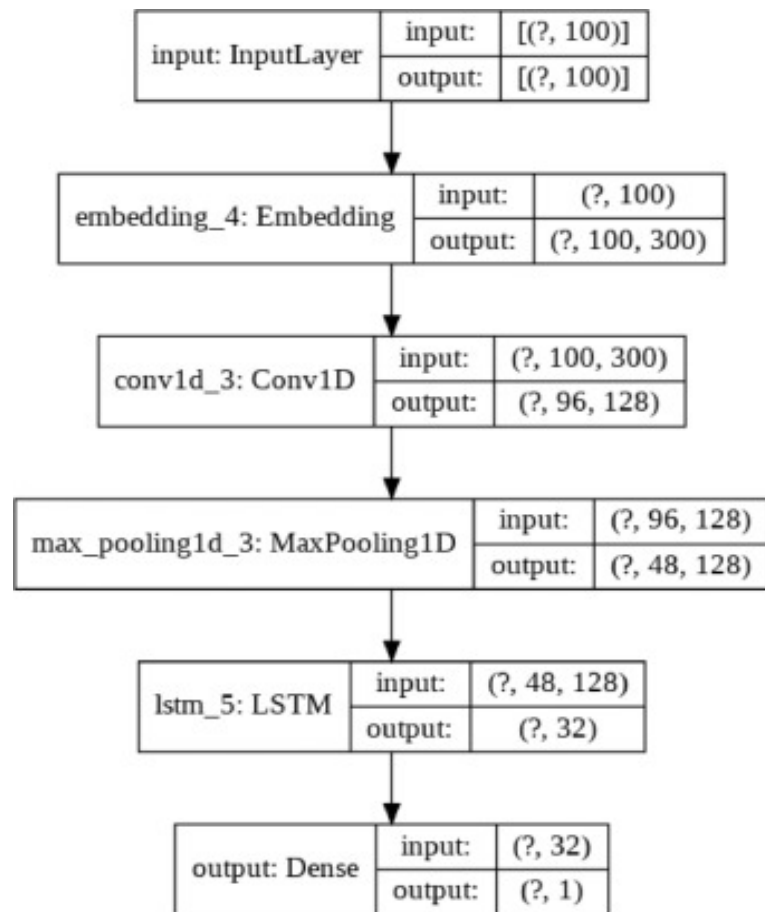
6. Lớp Dropout (tùy chọn):

- Mục đích: Giảm nguy cơ quá mức học và cải thiện khả năng tổng quát hóa.
- Ngẫu nhiên tắt một số nơ-ron trong quá trình huấn luyện để mô hình không phụ thuộc quá mức vào một số đặc trưng cụ thể.

7. Lớp Output (Sigmoid):

- Mục đích: Xuất ra xác suất của mỗi lớp (giả mạo hoặc không giả mạo).

- Sử dụng hàm kích hoạt Sigmoid (nếu chỉ có hai lớp) hoặc hàm Softmax (nếu có nhiều hơn hai lớp) để chuyển đổi đầu ra thành xác suất. Giá trị cao nhất trong các xác suất này là lớp dự đoán.



Hình 1.5: Mô hình CNN

*Nguồn: Proposed CharCNN encoder model

Chương 2

Mô hình thực nghiệm

Trong phần này, chúng tôi sẽ trình bày chương trình thực nghiệm của dự án. Bao gồm các bước dọn "*đẹp dữ*" liệu đầu vào, tiền xử lý dữ liệu thô, nhúng từ (Word embeddings) và cuối cùng là khởi chạy mô hình CNN.

Vì toàn bộ chương trình được viết bằng ngôn ngữ python nên chúng ta cần import các thư viện liên quan.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import re
import string
```

2.1 Chuẩn bị dữ liệu đầu vào

Bộ dữ liệu đầu vào của các bài báo True và Fake được lấy từ link:

<https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/fake-news-detection-datasets>

```
#ISOT's Data
#link about data: https://onlineacademiccommunity.uvic.ca/isot
  /2022/11/27/fake-news-detection-datasets/

df_fake = pd.read_csv("/content/gdrive/MyDrive/Colab-Notebooks
/Fake.csv")
df_true = pd.read_csv("/content/gdrive/MyDrive/Colab-Notebooks
/True.csv")
```

2.2 Dọn dẹp dữ liệu

Chúng ta loại bỏ các từ dừng (the, is, at, which, on, ...) và các ký tự , đường link, các ký tự xuống dòng, loại bỏ từ chứa con số ... để chuyển về chữ thường.

```
#Remove the stopword
def wordopt(text):
    text = text.lower()
    text = re.sub('[.*?\\]', '', text)
    text = re.sub("\\W", " ", text)
    text = re.sub('https?://\\S+|www\\.\\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\\n', '', text)
    text = re.sub('\\w*\\d\\w*', '', text)
    return text
```

2.3 Tiền xử lý dữ liệu và nhúng từ

Thực hiện việc chuyển đổi văn bản thành chuỗi số (index) sử dụng Tokenizer:

```
# Convert text to sequence of indices

vocab_size = 30000 # vocabulary size
embedding_dim = 300 # embedding dimension
max_len = 1000 # maximum length
tokenizer = Tokenizer(num_words=vocab_size, oov_token="<OOV>") # embed
    vocabulary tokens
tokenizer.fit_on_texts(train_sentences)
```

```
# Get the list of words and corresponding integer indices from the
tokenizer

word_index_items = list(tokenizer.word_index.items())

# Check if there are at least 201 words in the tokenizer
print(len(word_index_items))
if len(word_index_items) >= 201:
    # Get the 201st word in the list
    word_201, index_201 = word_index_items[200]

    print(f"The 201st word in the tokenizer is: '{word_201}', with
          integer index: {index_201}")
else:
    print("Tokenizer does not contain 201 words.")

tokenizer.word_index
```

Output:

```
{ '<OOV>': 1,
  'the': 2,
  'to': 3,
  'of': 4,
  'a': 5,
  'and': 6,
  'in': 7,
  's': 8,
  'that': 9,
  'on': 10,
  'for': 11,
  'is': 12,
  'trump': 13,
  'he': 14,
  'said': 15, ... }
```

2.4 Khởi tạo mô hình CNN

```
# CNN model
model = Sequential()

# Word embedding layer
```



```

model.add(Embedding(input_dim=vocab_size, output_dim=embedding_dim,
                    input_length=max_len))

# Convolutional and pooling layers
model.add(Conv1D(filters=filters, kernel_size=3, activation='relu'))
model.add(GlobalMaxPooling1D())

# Fully Connected layer
model.add(Dense(128, activation='relu'))
model.add(Dropout(dropout_rate))

# Output layer
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['
    accuracy'])

model.summary()

```

- **Lớp đầu vào:** Dữ liệu đầu vào bao gồm các bài viết được tiêu chuẩn hóa có độ dài đều nhau là 1000 từ, mỗi từ được ánh xạ tới một số tương ứng.
- **Lớp nhúng:** Đầu vào được lấy từ đầu ra của lớp đầu vào. Trong lớp này, mỗi từ được nhúng để thể hiện nó dưới dạng vector 300 chiều. Do đó, mỗi bài viết được biểu diễn dưới dạng ma trận có kích thước 1000 x 300.
- **Các chập:** Lớp Conv1D được sử dụng để trích xuất các tính năng từ ma trận đầu ra của lớp Nhúng. Lớp Conv1D đều sử dụng các bộ lọc có kích thước 64, tương ứng với kích thước hạt nhân là 3 và sử dụng chức năng kích hoạt ReLU.
- **Lớp GlobalMaxPooling:** Lớp này nhằm mục đích giảm kích thước của dữ liệu và giữ lại các giá trị tính năng quan trọng nhất.
- **Các lớp được kết nối đầy đủ:** Lớp này đặt các ly vector phẳng thu được từ đầu ra của lớp GlobalMaxPooling với 128 nơ-ron sử dụng chức năng kích hoạt ReLU.
- **Lớp đầu ra:** Lớp đầu ra sử dụng các Lớp được kết nối đầy đủ với một nơ-ron duy nhất và hàm kích hoạt Sigmoid để phân loại kết quả thành hai lớp.

Sau đây là kết quả của hàm `model.summary()` :

Model: "sequential"

| Layer (type) | Output Shape | Param # |
|---|-------------------|---------|
| embedding (Embedding) | (None, 1000, 300) | 9000000 |
| conv1d (Conv1D) | (None, 998, 64) | 57664 |
| global_max_pooling1d (GlobalMaxPooling1D) | (None, 64) | 0 |
| dense (Dense) | (None, 128) | 8320 |
| dropout (Dropout) | (None, 128) | 0 |
| dense_1 (Dense) | (None, 1) | 129 |
| Total params: 9066113 (34.58 MB) | | |
| Trainable params: 9066113 (34.58 MB) | | |
| Non-trainable params: 0 (0.00 Byte) | | |

Tiếp tục , chúng ta huấn luyện mô hình :

```
#Training model
history = model.fit(padded_train_sequences, train_labels, batch_size =
                    512, epochs=5, validation_data=(padded_test_sequences, test_labels))
```

Kết quả:

```
Epoch 1/5
62/62 [=====] - 23s 264ms/step - loss: 0.2928 -
accuracy: 0.9082 - val_loss: 0.0245 - val_accuracy: 0.9925
Epoch 2/5
62/62 [=====] - 15s 242ms/step - loss: 0.0130 -
accuracy: 0.9964 - val_loss: 0.0076 - val_accuracy: 0.9979
Epoch 3/5
62/62 [=====] - 14s 230ms/step - loss: 0.0030 -
accuracy: 0.9994 - val_loss: 0.0052 - val_accuracy: 0.9989
Epoch 4/5
62/62 [=====] - 14s 230ms/step - loss: 0.0012 -
accuracy: 0.9999 - val_loss: 0.0047 - val_accuracy: 0.9989
```

```
Epoch 5/5  
62/62 [=====] - 13s 215ms/step - loss: 7.5435e  
-04 - accuracy: 0.9999 - val_loss: 0.0045 - val_accuracy: 0.9988
```

Sau khi kết quả thu nhận được ở trên, kết quả huấn luyện mô hình rất tích cực. Trong quá trình huấn luyện, Loss giảm dần từ 0.2568 ở epoch 1 xuống còn 6.2778e-04 ở epoch 5. Điều này cho thấy mô hình đang giảm thiểu mất mát trên dữ liệu huấn luyện, đồng nghĩa với việc cải thiện khả năng dự đoán của mô hình.

Trong quá trình huấn luyện, Độ chính xác trên dữ liệu huấn luyện và Độ chính xác trên dữ liệu kiểm thử tăng dần đạt được độ chính xác cao.

Việc đào tạo mô hình trong dự án này sử dụng bộ dữ liệu: *ISOT phát hiện tin giả bộ dữ liệu*. Bộ dữ liệu bao gồm 23.481 bài báo được dán nhãn là giả mạo và 21.417 bài báo được dán nhãn là thật, bao gồm nhiều lĩnh vực khác nhau với trọng tâm chính là chính trị, kinh tế, văn hóa, v.v.

Các thử nghiệm được thực hiện với sự hỗ trợ của các thư viện và công cụ, bao gồm TensorFlow, NLTK, pandas và scikit-learn, hỗ trợ môi trường thử nghiệm được cung cấp bởi Google Colab.

Mô hình được xây dựng gồm CNN. Mô hình được đào tạo bằng cách sử dụng hàm tối ưu hóa "adam" và hàm mất mát được sử dụng là "binary_crossentropy" do các mô hình chỉ phân loại bài viết thành hai loại.

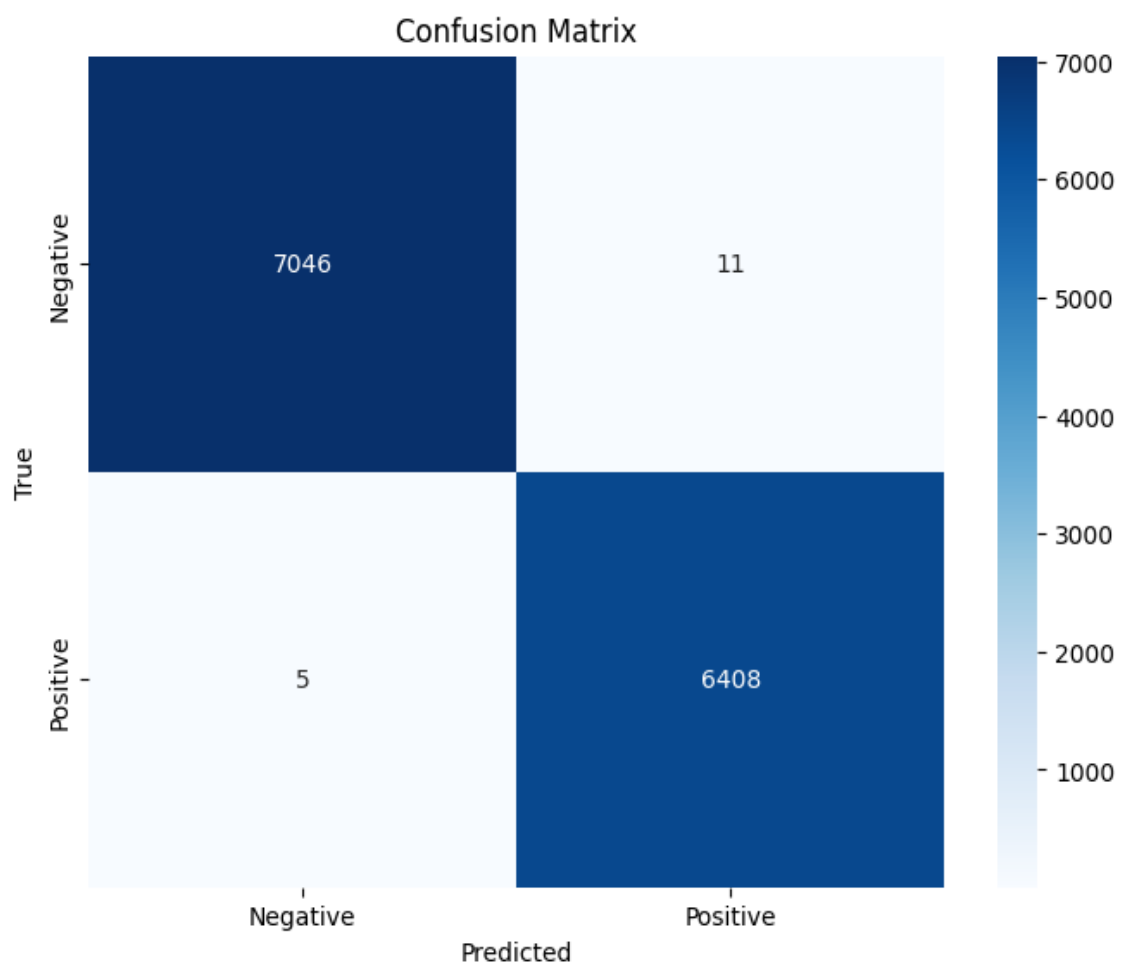
Chương 3

Kết quả mô hình thực nghiệm

Biểu đồ bên dưới minh họa các giá trị độ chính xác trong đào tạo và xác thực tương ứng của mô hình CNN.



Và đây là các ma trận nhầm lẫn cho từng mô hình đối với tập dữ liệu thử nghiệm.



Chương 4

Đánh giá và Thảo Luận

Chủ đề tập trung vào việc sử dụng các mô hình CNN, để xây dựng một mô hình phân loại để phân biệt giữa tin tức thật và tin giả. Nhìn chung, mô hình CNN. Mô hình đã cho kết quả rất thỏa đáng trên tập dữ liệu ISOT, đạt được một tỷ lệ độ chính xác vượt quá 99%.

Tuy nhiên, điều quan trọng cần lưu ý là các mô hình được phát triển trong nghiên cứu này chỉ được thử nghiệm trên các tập dữ liệu cụ thể. Để có thể xử lý hiệu quả một lượng lớn tin tức và được cập nhật thường xuyên, cần phải phát triển và cải tiến thêm các mô hình học tập sâu. Đây là một thách thức lớn đòi hỏi các phương pháp giải quyết thích hợp.

Phụ lục: Mã nguồn

Mã nguồn của dự án được xây dựng bằng ngôn ngữ Python và triển khai trong môi trường Google Colab.

Chương trình sử dụng các thư viện phổ biến như Pandas, NumPy, và Seaborn để xử lý tập dữ liệu đầu vào, bao gồm các bài báo thu thập từ một nguồn có sẵn. Sau khi hoàn tất quá trình xử lý dữ liệu, mô hình được đào tạo bằng các thuật toán học máy thông qua thư viện Scikit-learn (sklearn).

Để hiểu rõ chi tiết các bước xử lý và hoạt động cụ thể của chương trình, mã nguồn đầy đủ được cung cấp ở trang tiếp theo.

x

```
[1]: #connect colab to drive
from google.colab import drive
drive.mount('/content/gdrive')
```

Mounted at /content/gdrive

```
[ ]: from google.colab import drive
drive.mount('/content/drive')
```

```
[2]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import re
import string
```

```
[5]: #ISOT data's
#link of data: https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/fake-news-detection-datasets/

df_fake = pd.read_csv("/content/gdrive/MyDrive/Colab-Notebooks/Fake.csv")
df_true = pd.read_csv("/content/gdrive/MyDrive/Colab-Notebooks/True.csv")
```

```
[6]: df_fake["class"] = 0
df_true["class"] = 1
df_fake.shape, df_true.shape
```

```
[6]: ((23481, 5), (21417, 5))
```

```
[7]: df_merge = pd.concat([df_fake, df_true], axis =0 )
df_merge.head(10)
```

```
[7]:
```

| | title \ | text | subject \ |
|-----|---|-------|-----------|
| 0 | Donald Trump Sends Out Embarrassing New Year'... | | |
| 1 | Drunk Bragging Trump Staffer Started Russian ... | | |
| 2 | Sheriff David Clarke Becomes An Internet Joke... | | |
| ... | | | |
| | | | |
| 0 | Donald Trump just couldn t wish all Americans ... | | News |
| 1 | House Intelligence Committee Chairman Devin Nu... | | News |
| 2 | On Friday, it was revealed that former Milwauk... | | news |
| ... | | | |
| | | | |
| | date | class | |


```

0  December 31, 2017      0
1  December 31, 2017      0
2  December 30, 2017      0
...

```

```
[8]: df_merge.columns
```

```
[8]: Index(['title', 'text', 'subject', 'date', 'class'], dtype='object')
```

```
[9]: #remove unnecessary columns, keep only text and layers
df = df_merge.drop(["title", "subject", "date"], axis = 1)
```

```
[10]: df.columns
```

```
[10]: Index(['text', 'class'], dtype='object')
```

```
[11]: #shuffle data sets randomly
df = df.sample(frac = 1)
```

```
[12]: df.head()
```

```
[12]:
```

| | text | class |
|-------|---|-------|
| 3352 | WASHINGTON (Reuters) - Two top U.S. intelligen... | 1 |
| 10864 | WASHINGTON (Reuters) - Republican George W. Bu... | 1 |
| 13374 | KIEV (Reuters) - Ukrainian President Petro Por... | 1 |
| 23061 | Join Patrick every Wednesday at Independent Ta... | 0 |
| 10534 | CNN employees have resigned for their role in ... | 0 |

```
[13]: df.reset_index(inplace = True)
df.drop(["index"], axis = 1, inplace = True)
df.head()
```

```
[13]:
```

| | text | class |
|---|---|-------|
| 0 | WASHINGTON (Reuters) - Two top U.S. intelligen... | 1 |
| 1 | WASHINGTON (Reuters) - Republican George W. Bu... | 1 |
| 2 | KIEV (Reuters) - Ukrainian President Petro Por... | 1 |
| 3 | Join Patrick every Wednesday at Independent Ta... | 0 |
| 4 | CNN employees have resigned for their role in ... | 0 |

```
[14]: #Remove stop words
def wordopt(text):
    text = text.lower()
    text = re.sub('[\.*?\\]', '', text)
    text = re.sub("\\W", "", text)
    text = re.sub('https?:\\/\\S+|www\\.\\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\\n', '', text)
```

```
text = re.sub('\w*\d\w*', '', text)
return text
```

```
[15]: type(df["text"])
```

```
[15]: pandas.core.series.Series
```

```
[16]: df["text"] = df["text"].apply(wordopt)
```

```
[17]: x = df["text"]
      y = df["class"]
      x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3)
```

```
[18]: type(x_train)
```

```
[18]: pandas.core.series.Series
```

```
[19]: import numpy as np

      train_sentences = x_train.to_numpy()
      train_labels = y_train.to_numpy()

      test_sentences = x_test.to_numpy()
      test_labels = y_test.to_numpy()
```

```
[20]: #Create vocabulary to input train and test labels
      int32 = np.int32
      train_labels = train_labels.astype(int32)
      test_labels = test_labels.astype(int32)
```

```
[21]: #Verify the correctness of the train
      train_labels[0]
```

```
[21]: 1
```

```
[22]: from tensorflow.keras.preprocessing.text import Tokenizer
      from tensorflow.keras.preprocessing.sequence import pad_sequences
```

```
[23]: # Convert text to numeric string (index)

      vocab_size = 30000 #size
      embedding_dim = 300 #dimentions
      max_len = 1000 #length
      tokenizer = Tokenizer(num_words=vocab_size, oov_token="<OOV>") #embedding token_
      ↪ vocabulary
      tokenizer.fit_on_texts(train_sentences)
```

```
[24]: num=len(tokenizer.word_index)
      print(num) #print out the number of sentences from the data set
```

92827

```
[ ]: # Get the list of words and corresponding integers from the tokenizer
      word_index_items = list(tokenizer.word_index.items())

      # Check if there are at least 201 words in the tokenizer
      print(len(word_index_items))
      if len(word_index_items) >= 201:
          # Taken from number 201 in the list
          word_201, index_201 = word_index_items[200]

          print(f"The 201st word in the tokenizer is: '{word_201}', whose integer is:␣
↪{index_201}")
      else:
          print("Tokenizer does not have 201 words.")
```

```
[26]: tokenizer.word_index
```

```
[26]: {'<OOV>': 1,
      'the': 2,
      'to': 3,
      ....
      'have': 24,
      'not': 25,
      ...}
      'from': 26,
```

```
[27]: #Check if the document has been processed
      new_texts = ["elections doing run robotterror"]
      new_sequences = tokenizer.texts_to_sequences(new_texts)
      print(new_sequences)
```

[[589, 422, 418, 1]]

```
[ ]: from itertools import islice
      from collections import Counter

      # Number of words in dictionary
      num_words_in_dictionary = len(tokenizer.word_index)

      # Calculate frequency of each word in training data
      word_frequencies = Counter(" ".join(train_sentences).split())

      # Print dictionary with frequency from 20000 to 21000
      start_index, end_index = 91900, 92000
```

```

for word, index in islice(tokenizer.word_index.items(), start_index, end_index):
    frequency = word_frequencies[word] if word in word_frequencies else 0
    print(f"{word}: {index} - Frequency: {frequency}")

```

```

[29]: # Get the data processed in the above step into the train set
train_sequences = tokenizer.texts_to_sequences(train_sentences)
padded_train_sequences = pad_sequences(train_sequences, maxlen = max_len,
    ↪truncating='post', padding = 'post')

```

```

[30]: # Get the data processed in the above step into the test set
test_sequences = tokenizer.texts_to_sequences(test_sentences)
padded_test_sequences = pad_sequences(test_sequences, maxlen = max_len,
    ↪truncating='post', padding = 'post')

```

```

[31]: from keras.models import Sequential
from keras.layers import Embedding, Conv1D, MaxPooling1D, LSTM, Dense, Dropout,
    ↪GlobalMaxPooling1D, SimpleRNN, Bidirectional, concatenate

```

```

[32]: # constants in the model
filters = 64
kernel_size = 3
pool_size = 2
lstm_units = 128
rnn_units = 64
dropout_rate = 0.5
num_filters = 32

```

```

[ ]: # CNN model
model = Sequential()

# Embedding layer from
model.add(Embedding(input_dim=vocab_size, output_dim=embedding_dim,
    ↪input_length=max_len))

# Convolutional and pooling layers
model.add(Conv1D(filters=filters, kernel_size=3, activation='relu'))
model.add(GlobalMaxPooling1D())

# Fully Connected layer
model.add(Dense(128, activation='relu'))
model.add(Dropout(dropout_rate))

# Output layer
model.add(Dense(1, activation='sigmoid'))

# Compile model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

```

```
model.summary()
```

```
[34]: #Training model
history = model.fit(padded_train_sequences, train_labels, batch_size = 512,
                    epochs=5, validation_data=(padded_test_sequences, test_labels))
```

```
Epoch 1/5
62/62 27s 297ms/step -
accuracy: 0.8509 - loss: 0.3868 - val_accuracy: 0.9969 - val_loss: 0.0123
Epoch 2/5
62/62 11s 169ms/step -
accuracy: 0.9947 - loss: 0.0207 - val_accuracy: 0.9978 - val_loss: 0.0066
Epoch 3/5
62/62 21s 171ms/step -
accuracy: 0.9982 - loss: 0.0080 - val_accuracy: 0.9991 - val_loss: 0.0045
Epoch 4/5
62/62 21s 172ms/step -
accuracy: 0.9991 - loss: 0.0039 - val_accuracy: 0.9989 - val_loss: 0.0041
Epoch 5/5
62/62 20s 171ms/step -
accuracy: 0.9997 - loss: 0.0014 - val_accuracy: 0.9990 - val_loss: 0.0038
```

```
[35]: model.save('/content/gdrive/MyDrive/Colab Notebooks/FakeNewDetectionCNN.h5')
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or
`keras.saving.save_model(model)`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')` or `keras.saving.save_model(model,
'my_model.keras')`.
```

```
[ ]: import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))

acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']

plt.figure(figsize=(8, 8))
plt.subplot(2, 1, 1)
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.ylabel('Accuracy')
plt.ylim([min(plt.ylim()), 1])
```

```
plt.title('Training and Validation Accuracy')

plt.subplot(2, 1, 2)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.legend (loc='upper right')
plt.ylabel('Cross Entropy')
plt.title( 'Training and Validation Loss')
plt.xlabel('epoch')
plt.show()
```

```
[37]: # Predictions on new data - REAL NEWS
```

```

new_texts = ["BRUSSELS (Reuters) - NATO allies on Tuesday welcomed President
↳Donald Trump s decision to commit more forces to Afghanistan, as part of a new
↳U.S. strategy he said would require more troops and funding from America s
↳partners. Having run for the White House last year on a pledge to withdraw
↳quickly from Afghanistan, Trump reversed course on Monday and promised a
↳stepped-up military campaign against Taliban insurgents, saying: Our troops
↳will fight to win send about 4,000 more U.S. troops to add to the roughly
↳8,400 now deployed in Afghanistan. But his speech did not define benchmarks
↳for successfully ending the war that began with the U.S.-led invasion of
↳Afghanistan in 2001, and which he acknowledged had required an extraordinary
↳sacrifice of blood and treasure . We will ask our NATO allies and global
↳partners to support our new strategy, with additional troops and funding
↳increases in line with our own. That comment signaled he would further
↳increase pressure on U.S. partners who have already been jolted by his
↳repeated demands to step up their contributions to NATO and his description of
↳the alliance as obsolete - even though, since taking office, he has said this
↳is no longer the case. NATO Secretary General Jens Stoltenberg said in a
↳statement: NATO remains fully committed to Afghanistan and I am looking
↳forward to discuss the way ahead with (Defense) Secretary (James) Mattis and
↳our Allies and international partners. NATO has 12,000 troops in Afghanistan,
↳and 15 countries have pledged more, Stoltenberg said. Britain, a leading NATO
↳member, called the U.S. commitment very welcome . In my call with Secretary
↳Mattis yesterday we agreed that despite the challenges, we have to stay the
↳course in Afghanistan to help build up its fragile democracy and reduce the
↳terrorist threat to the West, Defense Secretary Michael Fallon said. Germany,
↳which has borne the brunt of Trump's criticism over the scale of its defense
↳spending, also welcomed the new U.S. plan. plan. Our continued commitment is
↳necessary on the path to stabilizing the country, a government consolidated
↳said. In June, European allies had already pledged more troops but had not
↳given details on numbers, waiting for the Trump administration to outline its
↳strategy for the region.Nearly 16 years after the U.S.-led invasion - a
↳response to the Sept. 11 attacks which were planned by al Qaeda leader Osama
↳bin Laden from Afghanistan - the country is still struggling with weak central
↳government and a Taliban insurgency. Trump said he shared the frustration of
↳the American people who were weary of war without victory, but a hasty
↳withdrawal would create a vacuum for groups like Islamic State and al Qaeda to
↳fill.""]

new_sequences = tokenizer.texts_to_sequences(new_texts)
new_padded_sequences = pad_sequences(new_sequences, maxlen=max_len,
↳padding='post')

predictions = model.predict(new_padded_sequences)
print("Predictions:", predictions)

```

```

1/1 0s 425ms/step
Predictions: [[0.99998415]]

```

[38]: *# Prediction on new data - FAKE NEWS*

```
new_texts = ["Vic Bishop Waking TimesOur reality is carefully constructed by
↳powerful corporate, political and special interest sources in order to
↳covertly sway public opinion. Blatant lies are often televised about
↳terrorism, food, war, health, etc. They are fashioned to sway Public opinion
↳and condition viewers to accept what has become destructive norms.The practice
↳of manipulating and controlling public opinion with amplified media messages
↳has become so common that there is a whole industry formed around this out how
↳to spin information to journalists, similar to the lobbying of government. It
↳is never really clear just how much the journalists receive because the news
↳industry has become complacent The messages that it presents are shaped by
↳corporate powers who often spend millions on advertising the six conglomerates
↳that own 90% of the media: General Electric (GE), News-Corp, Disney, Viacom,
↳Time Warner, and CBS. Yet, these corporations function under many different
↳brands, such as FOX, ABC, CNN, Comcast, Wall Street Journal, etc, giving
↳people the perception of choice As Tavistock s researchers showed, it was
↳important that the victims of mass brainwashing not be aware that their
↳environment is being controlled; there should thus be a vast number of sources
↳for information, whose messages could be varied slightly, so as to mask the
↳sense of external control. ~ Specialist of mass brainwashing, L. WolfeNew
↳Brainwashing Tactic Called AstroturfWith alternative media on the rise, the
↳propaganda machine continues to expand. Below is a video of Sharyl Attkisson,
↳investigative reporter with CBS, during which she explains how astroturf, or
↳fake groundswell movements, are used to spin information not only to influence
↳journalists but to sway public opinion. Astroturf is a perversion of
↳groundswells. Astroturf is when political, corporate or other special
↳interests disguise themselves and publish blogs, start facebook and twitter
↳accounts, publish ads, letters to the editor, or simply post comments online,
↳to try to fool you into thinking an independent or grassroots movement is
↳speaking . ~ Sharyl Attkisson, Investigative ReporterHow do you separate facts
↳from fiction? Sharyl Attkisson finishes her talk with some insights on how to
↳identify signs of propaganda and astroturfing These methods are used to give
↳people the impression that there is widespread support for an agenda, when, in
↳reality, one may not exist. Astroturf tactics are also used to discredit or
↳criticize those that disagree with certain agendas, using stereotypical names
↳such as conspiracy theorist or quack. When in fact when someone dares to
↳reveal the truth or questions the official story, it should spark a deeper
↳curiosity and encourage further scrutiny of the information.This article
↳(Journalist Reveals Tactics Brainwashing Industry Uses to Manipulate the
↳Public) was originally created and published by Waking Times and is published
↳here under a Creative Commons license with attribution to Vic Bishop and
↳WakingTimes.com. It may be re-posted freely with proper attribution, author
↳bio, and this copyright statement. READ MORE MSM PROPAGANDA NEWS AT: 21st
↳Century Wire MSM Watch Files"]

new_sequences = tokenizer.texts_to_sequences(new_texts)
new_padded_sequences = pad_sequences(new_sequences, maxlen=max_len,
↳padding='post')
```



```
predictions = model.predict(new_padded_sequences)
print("Predictions:", predictions)
```

1/1 0s 26ms/step

Predictions: [[1.8782396e-07]]

```
[39]: # Test prediction
new_texts = ["Question dr ron paul outer limit radio dr paul serve twelve term u
↳house representative threetime candidate u president devote political career
↳defense individual liberty sound money noninterventionist foreign policy judge
↳andrew napolitano call thomas jefferson day serve flight surgeon u air force
↳dr paul move texas begin civilian medical practice deliver four thousand baby
↳career obstetrician dr paul serve congress carol paul wife fifty year five
↳child many grandchild greatgrandchildren ron paul new york post write
↳politician buy special interest people public life thick thin rain shine stick
↳principles added parliamentary colleague ron paul one ron paul never vote
↳legislation unless proposes measure explicitly authorized constitution also
↳congressman ron paul never vote raise tax ron paul never vote unbalanced
↳budget ron paul never vote federal restriction gun ownership ron paul never
↳vote raise congressional pay ron paul never take governmentpaid junket ron
↳paul never vote increase power executive branch ron paul vote patriot act ron
↳paul vote regulate internet ron paul vote iraq war ron paul participated
↳lucratively congressional pension program ron paul chairman ron paul institute
↳peace prosperity nonprofit educational charity host daily ron paul liberty
↳report special thanks daniel mcadams chris rossini dylan charles chris duane
↳mp audio link please check dr paul late book revolution ten year learn out
↳limit radio website"]
new_sequences = tokenizer.texts_to_sequences(new_texts)
new_padded_sequences = pad_sequences(new_sequences, maxlen=max_len,
↳padding='post')

predictions = model.predict(new_padded_sequences)
print("Predictions:", predictions)
```

1/1 0s 17ms/step

Predictions: [[0.01349811]]

```
[40]: # Test prediction
```

```

new_texts = ["Suge knight claim tupac still alive twentyone year ago legendary u
↳rapper tupac amaru shakur die shot street la vega september since conspiracy
↳theory image repeatedly emerge prove rapper still alive man sit car tupac shot
↳fire speaks make bizarre story year interview american tv station fox music
↳mogul suge explain life imprisonment murder explain former best friend live
↳anymore suge knight interview leave hospital pac laugh make joke ca nt
↳understand someone state health change good bad ask believe rapper still alive
↳suge reply tell never know pac people believe death incidentally music mogul
↳alone opinion tupac shakur could still alive former policeman recently claimed
↳pay million euro fake tupac death official put follow word record world need
↳to know do shame exchange honesty money ca nt die without world knowing
↳include popular story tupac tupac life cuba godmother political activist
↳assata olugbala shakur suge knight kill tupac tupac back rapper kasinova tha
↳spread music p diddy notorious big order tupac murder tupac member illuminati
↳illuminati kill tupac become powerful tupac abducted alien fbi kill tupac
↳related article suge knight finally admit tupac alive video elvis presley
↳alive people convince elvis alive new photo emerge"]
new_sequences = tokenizer.texts_to_sequences(new_texts)
new_padded_sequences = pad_sequences(new_sequences, maxlen=max_len,
↳padding='post')

predictions = model.predict(new_padded_sequences)
print("Predictions:", predictions)

```

1/1 0s 17ms/step

Predictions: [[0.00018336]]

[41]: # Test prediction

```

new_texts = ["Senate Majority Whip John Cornyn (R-TX) thought it would be a good
↳idea to attack Special Counsel Robert Mueller over the Russia probe. As
↳Mueller s noose tightens, Republicans are losing their sh-t and attacking
↳Mueller and the FBI in order to protect probably the most corrupt president
↳ever.Former Attorney General Eric Holder tweeted on Friday, Speaking on behalf
↳of the vast majority of the American people, Republicans in Congress be
↳forewarned: any attempt to remove Bob Mueller will not be tolerated. Cornyn
↳retweeted Holder to say, You don't. You don't https://t.co/7lHYkIloyz Senator
↳JohnCornyn (@JohnCornyn) December 16, 2017Bloomberg's Steven Dennis tweeted on
↳Saturday that [Cornyn]'s beef is with Holder, not Mueller, but Cornyn
↳responded to say, But Mueller needs to clean house of partisans. But Mueller
↳needs to clean house of partisans https://t.co/g8SwgAKtfH Senator JohnCornyn
↳(@JohnCornyn) December 16, 2017The Washington Post s Greg Sargent asked
↳Cornyn, Will you accept the findings of the Mueller probe as legitimate,
↳@JohnCornyn? Makes sense to me to wait to see what they are first, Cornyn
↳responded.Makes sense to me to wait to see what they are first https://t.co/
↳9lCqpYujKN Senator JohnCornyn (@JohnCornyn) December 16, 2017Republicans are
↳trying to discredit Mueller and Twitter users took notice.If you even THINK of
↳firing Mueller I ll make it my life's mission to make sure this is your last
↳term, buddy. Mrs. SMH (@MRSSMH2) December 16, 2017Carrollton, TX here Ready
↳and willing to help get Cruz and Cornyn out Jules012 (@JulesLorey1) December
↳16, 2017Garland, TX here Same! Bye Bye TDK (@ejkmom1998) December 16,
↳2017Austin, TX. #IStandWithMueller. Cronyn is a fake representative. He
↳represents his own interests and everything to profit himself Vj (@Tex92eye)
↳December 16, 2017I stand with Mueller! Kenneth Shipp (@shipp_kenneth) December
↳16, 2017He speaks for me Its BS how you cover up for a Russia pawn If Trump
↳not gulty why would Mueller be fired to cover up Ellen Reeher Morris
↳(@EllenMorris1222) December 16, 2017@EricHolder speaks for 69% of Americans
↳according to recent polling. That's the vast majority in my book. You were
↳around for the Saturday Night Massacre @JohnCornyn. Firing Mueller will be
↳X100! Lori Winters (@LoriW66) December 16, 2017Country over party. pic.twitter.
↳com/NXEX9rGBgu PittieBoo (@PittieBoo) December 16, 2017He speaks for me,
↳@JohnCornyn , and he speaks for the vast majority of American citizens who,
↳you should remember, vote. See you in 2018. Andrew Silver (@standsagreenoak)
↳December 16, 2017I might just move to Texas to get those cronies tossed out of
↳office. Blue wave is coming for the corrupt. Ollie (@marciebp) December 16,
↳2017Good try, John. History will not be kind to you.Photo by Ann Heisenfelt/
↳Getty Images"]

new_sequences = tokenizer.texts_to_sequences(new_texts)
new_padded_sequences = pad_sequences(new_sequences, maxlen=max_len,
↳padding='post')

predictions = model.predict(new_padded_sequences)
print("Predictions:", predictions)

```

1/1 0s 17ms/step

Predictions: [[2.8845284e-07]]

```
[42]: type(padded_train_sequences[0])
```

```
[42]: numpy.ndarray
```

```
[43]: print()
```

```
[44]: import matplotlib.pyplot as plt
from ipywidgets import widgets, Layout, VBox
from IPython.display import display, clear_output
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity

# Function to check articles
def record_tra_bai_bao(text):
    new_texts = [text]
    new_sequences = tokenizer.texts_to_sequences(new_texts)
    new_padded_sequences = pad_sequences(new_sequences, maxlen=max_len,
    ↪padding='post')

    predictions = model.predict(new_padded_sequences)
    print("Predictions:", predictions)
    if predictions[0][0] > 0.5:
        confidence = predictions[0][0] * 100
        return f"The above information is true ({confidence:.2f}%)"
    else:
        confidence = ( predictions[0][0]) * 100
        return f"The above information is false ({confidence:.2f}%)"

# Create text input field
text_input = widgets.Text(placeholder='Enter text article',
    ↪layout=Layout(width='70%', margin='0 auto 10px auto'))

# Create "Check article" button
check_button = widgets.Button(description='Check article', button_style=
    ↪'success', layout=Layout(width='20%', margin='0 auto 10px auto'))

# Create a label to display the results
result_label = widgets.HTML(value='', layout=Layout(margin= '20px'))

# Event handler when the button is pressed
def on_button_click(b):
    result = kiem_tra_bai_bao(text_input.value)
    result_label.value = result

# Draw a simple graph
```

```

fig, ax = plt.subplots()
ax.plot([1, 2, 3, 4], [10, 20, 25, 30], label='Sample data')
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_title('Simple chart')
ax.legend()

# Display chart in VBox
with out:
    clear_output(wait=True)
    plt.show()

# Bind event handler to button click event
check_button.on_click(on_button_click)

# Create VBox with rectangular border and white background
vbox = VBox([text_input, check_button, result_label], layout=
↳Layout(width='50%', margin='0 auto', border='2px solid #000',
↳background_color='#fff', align_items='center', justify_content='center'))

# Create output widget to display chart
out = widgets.Output(layout={'border': '1px solid black ', 'height': '300px',
↳'margin': '0 auto'})

# Add output widget to VBox
vbox.children += (out,)

# Set background color for entire VBox
vbox.layout.background_color = ' #fff'

# Display VBox
display(vbox)

```

Tài liệu tham khảo

- [1] I. K. Sastrawan, I. Bayupati và D. M. S. Arsa, “Phát hiện tin tức giả bằng phương pháp học sâu dựa trên CNN-RNN,” *CNTT chuyển phát nhanh*, tập 8, số 3, trang 396–408, 2022.

[Detection of fake news using deep learning CNN-RNN based methods - ScienceDirect](#)

- [2] J. A. Nasir, O. S. Khan và I. Varlamis, “Phát hiện tin tức giả: Phương pháp học sâu kết hợp giữa CNN-RNN,” *Tạp chí quốc tế về dữ liệu quản lý thông tin*, tập 1, số 1, p. 100007, 2021.

[Fake news detection: A hybrid CNN-RNN based deep learning approach - ScienceDirect](#)

Bộ dữ liệu đầu vào của các bài báo True và Fake được lấy từ link:

<https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/fake-news-detection-datasets>