

**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

|          |                                                                         |           |
|----------|-------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Giới Thiệu và cơ sở mô hình CNN</b>                                  | <b>2</b>  |
| 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         |
| <b>2</b> | <b>Mô hình thực nghiệm</b>                                              | <b>10</b> |
| 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        |
| <b>3</b> | <b>Kết quả mô hình thực nghiệm</b>                                      | <b>16</b> |
| <b>4</b> | <b>Đánh giá và Thảo Luận</b>                                            | <b>18</b> |

# 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

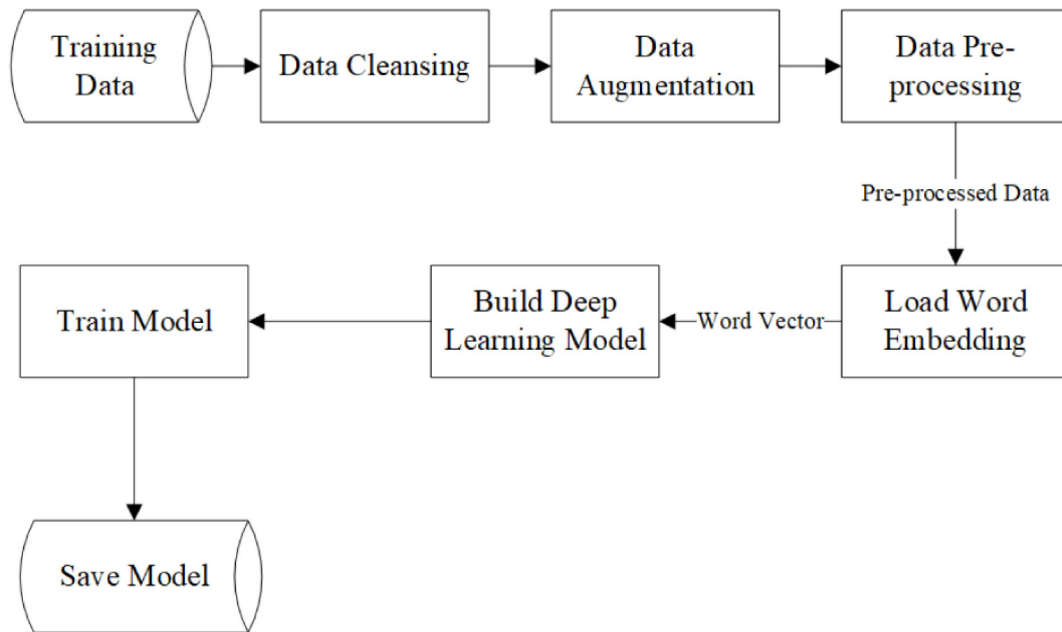
Mục tiêu chính của nghiên cứu là xây dựng một mô hình phân loại tự động có khả năng xác định các bài báo giả mạo hoặc thiếu độ tin cậy, sử dụng công nghệ CNN mới. Bằng cách này, nghiên cứu hướng đến việc phát triển một công cụ có hiệu quả trong việc giảm thiểu sự lan truyền của tin sai lệch, góp phần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

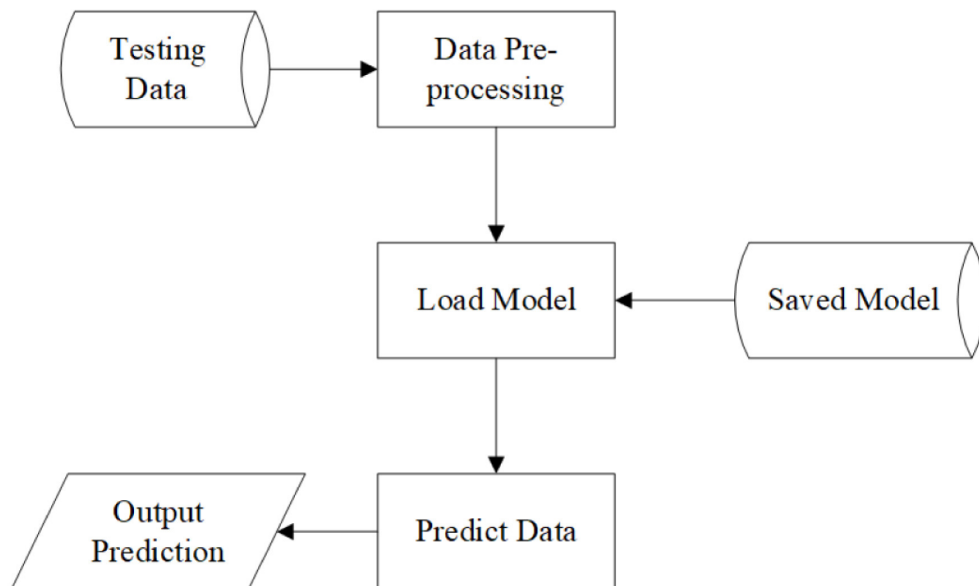
Dự án này trải qua một số giai đoạn, như được mô tả trong Hình 1.1 và 1.2. Nó bắt đầu bằng việc lấy dữ liệu huấn luyện từ cơ sở dữ liệu, sau đó là làm sạch dữ liệu thông qua việc loại bỏ các từ dừng và dấu chấm câu. Bước tiếp theo liên quan đến việc xử lý trước dữ liệu để chuyển đổi dữ liệu văn bản sang 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

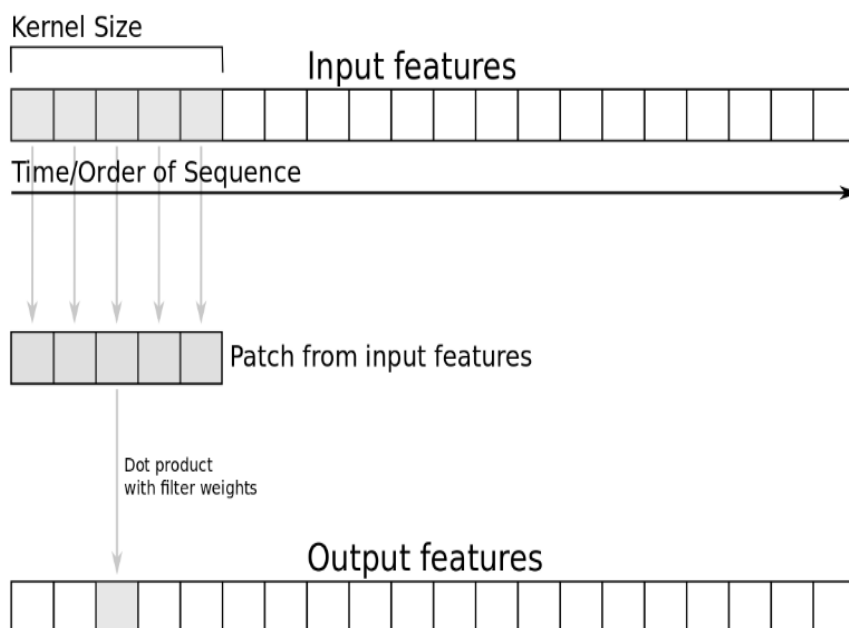
\*Nguồn: Tài liệu tham khảo số [1]

## 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.



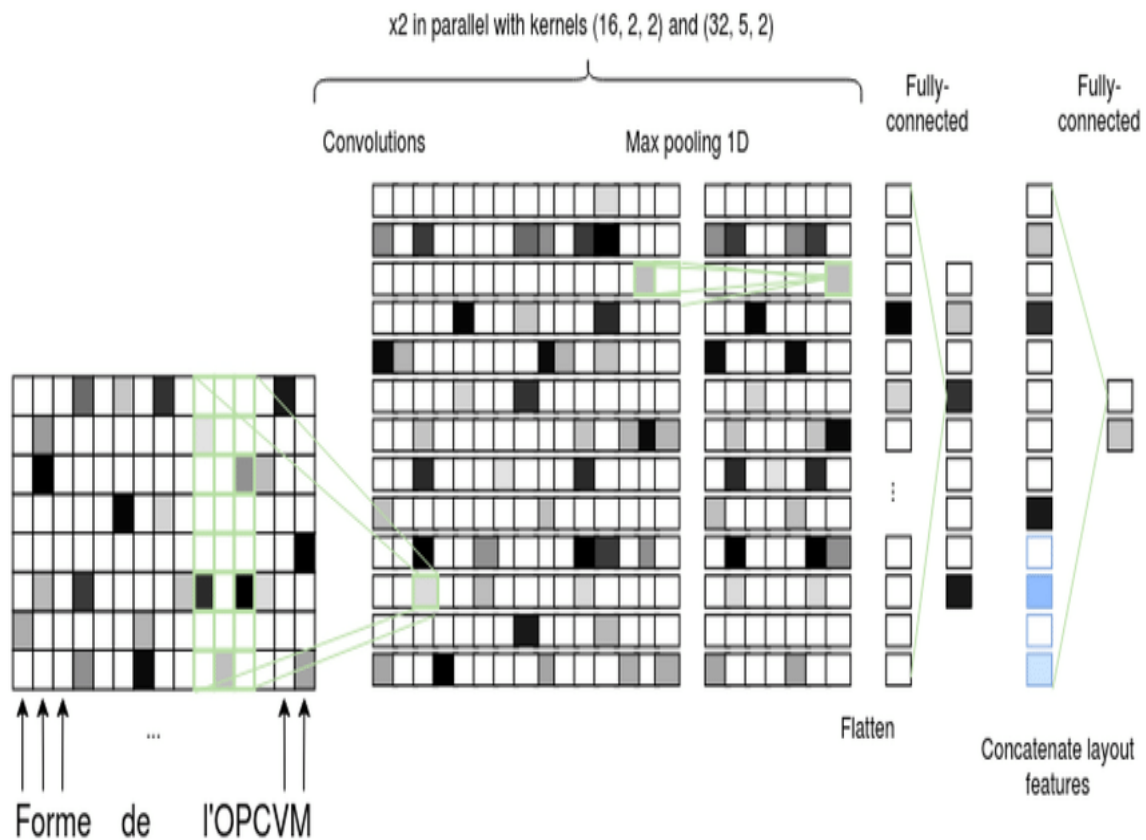
Hình 1.3: Mô phỏng mạng thần kinh tích chập

\*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.

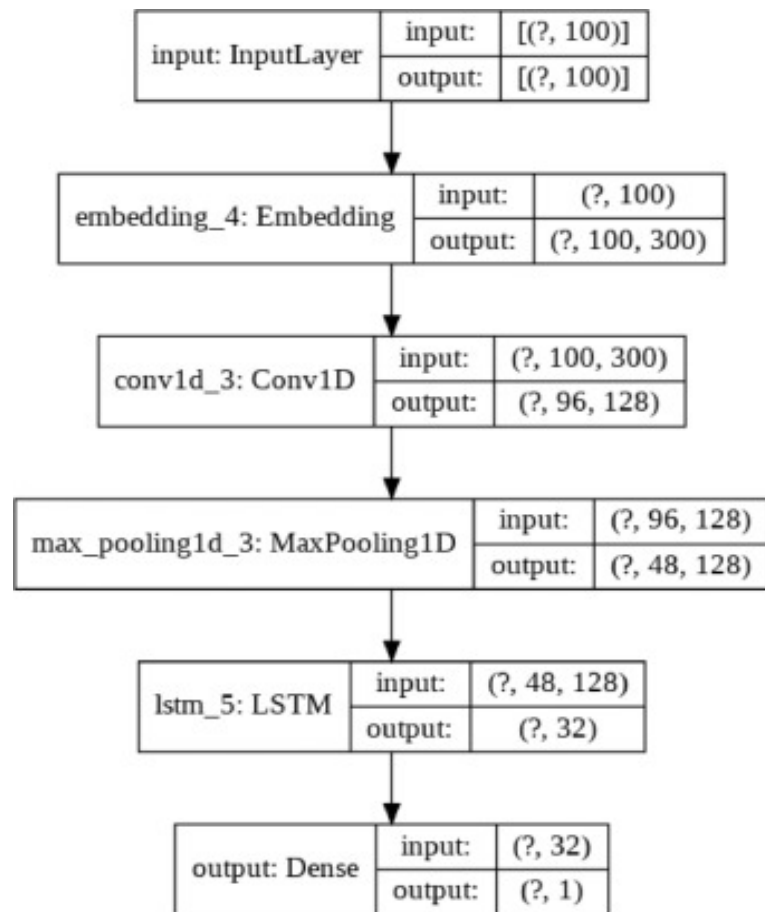
## 6. Lớp Dropout (tùy chọn):

- 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.

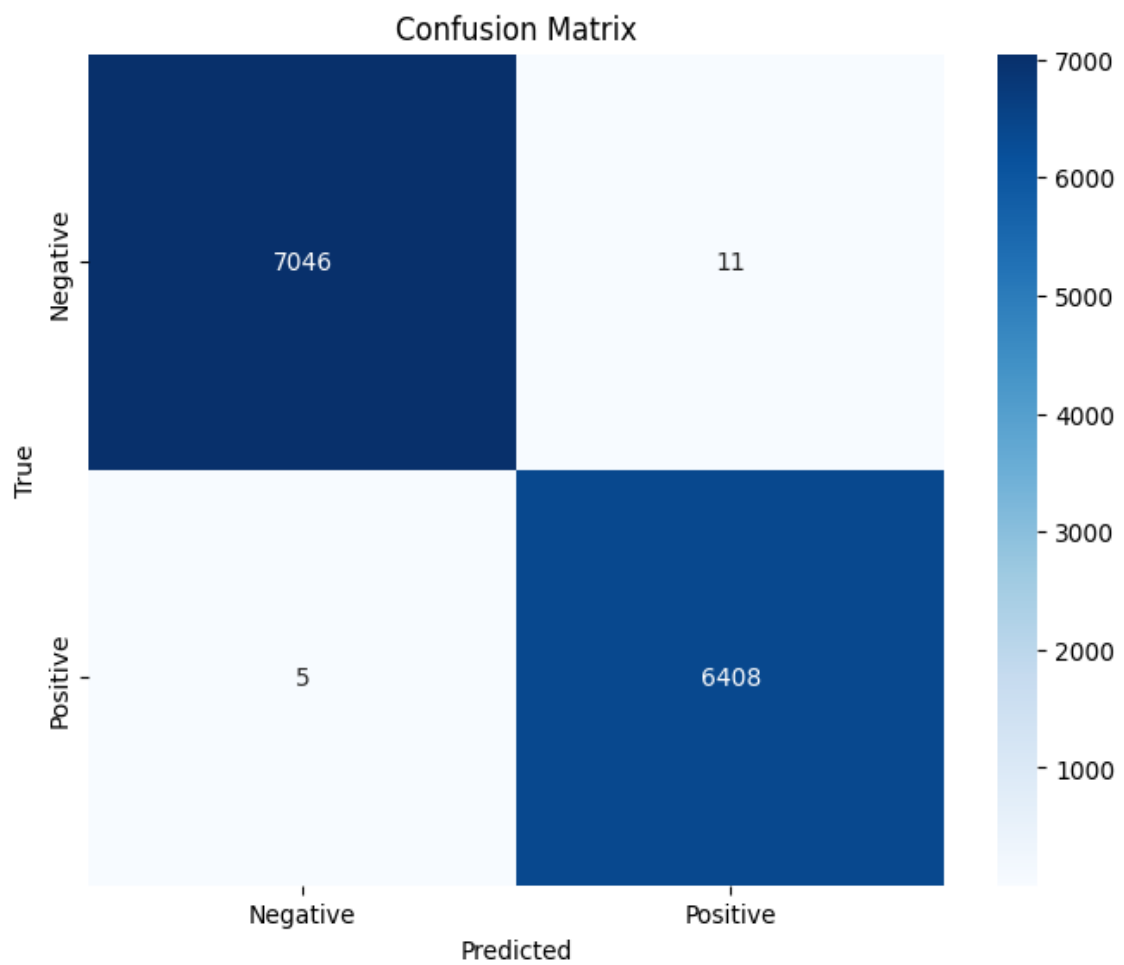
## Chương 3

### 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.

# 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>