BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**🙞 🕮 🙜**



**BÀI TẬP NHÓM**

**MÁY HỌC NÂNG CAO**

**ĐỀ TÀI**

**ỨNG DỤNG HỌC SÂU VÀO**

**BÀI TOÁN PHÂN LOẠI DỮ LIỆU**

**Giảng viên hướng dẫn:**

Phạm Nguyên Khang

**Sinh viên thực hiện:**

|  |  |
| --- | --- |
| Nguyễn Thành Hoàng Hải | B1812339 |
| Võ Ngọc Long | B1812282 |

Cần Thơ, 11/2021

# MỤC LỤC

[MỤC LỤC 1](#_Toc89180774)

[DANH MỤC HÌNH 3](#_Toc89180775)

[DANH MỤC BẢNG BIỂU 4](#_Toc89180776)

[MÔ TẢ YÊU CẦU BÀI TOÁN 5](#_Toc89180777)

[BÁO CÁO KẾT QUẢ 6](#_Toc89180778)

[I. Giới thiệu về dataset 6](#_Toc89180779)

[II. Các bước xây dựng mô hình 7](#_Toc89180780)

[1. Xây dựng hàm create\_dataset\_from\_dir 7](#_Toc89180781)

[2. Đọc dữ liệu 7](#_Toc89180782)

[3. Xây dựng mô hình 8](#_Toc89180783)

[III. Kết quả huấn luyện 10](#_Toc89180784)

[1. Kết quả mô hình 1 10](#_Toc89180785)

[1.1. Kết quả độ chính xác mô hình 10](#_Toc89180786)

[1.2. Kết quả độ lỗi mô hình 11](#_Toc89180787)

[1.3. Mô tả chi tiết phần dự đoán của mô hình 12](#_Toc89180788)

[2. Kết quả mô hình 2 12](#_Toc89180789)

[2.1. Kết quả độ chính xác mô hình 13](#_Toc89180790)

[2.2. Kết quả độ lỗi mô hình 13](#_Toc89180791)

[2.3. Mô tả chi tiết phần dự đoán của mô hình 14](#_Toc89180792)

[3. Kết quả mô hình 3 14](#_Toc89180793)

[3.1. Kết quả độ chính xác mô hình 15](#_Toc89180794)

[3.2. Kết quả độ lỗi mô hình 15](#_Toc89180795)

[3.3. Mô tả chi tiết phần dự đoán của mô hình 16](#_Toc89180796)

[4. Kết quả mô hình 4 16](#_Toc89180797)

[4.1. Kết quả độ chính xác mô hình 17](#_Toc89180798)

[4.2. Kết quả độ lỗi mô hình 17](#_Toc89180799)

[4.3. Mô tả chi tiết phần dự đoán của mô hình 18](#_Toc89180800)

[5. Kết quả mô hình thứ năm 18](#_Toc89180801)

[5.1. Kết quả độ chính xác mô hình 19](#_Toc89180802)

[5.2. Kết quả độ lỗi mô hình 19](#_Toc89180803)

[5.3. Mô tả chi tiết phần dự đoán của mô hình 20](#_Toc89180804)

[6. So sánh kết quả 21](#_Toc89180805)

[6.1. So sánh giữa mô hình 1, mô hình 2 và mô hình 3 21](#_Toc89180806)

[6.2. So sánh giữa mô hình 4 và mô hình 5 21](#_Toc89180807)

[6.3. So sánh giữa mô hình 3 và mô hình 4 22](#_Toc89180808)

[IV. Xây dựng ứng dụng 23](#_Toc89180809)

[KẾT QUẢ ĐẠT ĐƯỢC 24](#_Toc89180810)

[TÀI LIỆU THAM KHẢO 25](#_Toc89180811)

[PHỤ LỤC 26](#_Toc89180812)

# DANH MỤC HÌNH

[Hình 1: Biểu đồ minh họa độ chính xác của mô hình thứ nhất 10](#_Toc89180813)

[Hình 2: Biểu đồ minh họa độ lỗi của mô hình thứ nhất 11](#_Toc89180814)

[Hình 3: Mô tả chi tiết phần dự đoán của mô hình thứ nhất 12](#_Toc89180815)

[Hình 4: Biểu đồ minh họa độ chính xác của mô hình thứ hai 13](#_Toc89180816)

[Hình 5: Biểu đồ minh họa độ lỗi của mô hình thứ hai 13](#_Toc89180817)

[Hình 6: Mô tả chi tiết phần dự đoán của mô hình thứ hai 14](#_Toc89180818)

[Hình 7: Biểu đồ minh họa độ chính xác của mô hình thứ ba 15](#_Toc89180819)

[Hình 8: Biểu đồ minh họa độ lỗi của mô hình thứ ba 15](#_Toc89180820)

[Hình 9: Mô tả chi tiết phần dự đoán của mô hình thứ ba 16](#_Toc89180821)

[Hình 10: Biểu đồ minh họa độ chính xác của mô hình thứ tư 17](#_Toc89180822)

[Hình 11: Biểu đồ minh họa độ lỗi của mô hình thứ ba 17](#_Toc89180823)

[Hình 12: Mô tả chi tiết phần dự đoán của mô hình thứ tư 18](#_Toc89180824)

[Hình 13: Biểu đồ minh họa độ chính xác của mô hình thứ năm 19](#_Toc89180825)

[Hình 14: Biểu đồ minh họa độ lỗi của mô hình thứ năm 19](#_Toc89180826)

[Hình 15: Mô tả chi tiết phần dự đoán của mô hình thứ năm 20](#_Toc89180827)

[Hình 16: Giao diện ứng dụng 23](#_Toc89180828)

# DANH MỤC BẢNG BIỂU

[Bảng 1: Thông tin về tập dữ liệu VB2000 6](#_Toc89180829)

[Bảng 2: So sánh giữa 3 mô hình 21](#_Toc89180830)

[Bảng 3: So sánh giữa hai mô hình 3 và mô hình 4 21](#_Toc89180831)

[Bảng 4: So sánh giữa hai mô hình có kết quả huấn luyện tốt nhất 22](#_Toc89180832)

# MÔ TẢ YÊU CẦU BÀI TOÁN

Tập dữ liệu được sử dụng: Dữ liệu văn bản tiếng Việt.

Xây dựng các mô hình đáp ứng yêu cầu:

* Thử nghiệm với nhiều kiến trúc mạng khác nhau, so sánh hiệu quả phân nhãn của các kiến trúc mạng => chọn kiến trúc mạng tốt nhất trong số đó.
* Xây dựng một ứng dụng minh hoạ việc sử dụng mô hình đề xuất.
* Đầu vào: dữ liệu văn bản.
* Đầu ra: nhãn của dữ liệu.

# BÁO CÁO KẾT QUẢ

## Giới thiệu về dataset

Tập dữ liệu VB2000 gồm 2000 file văn bản chứa đoạn văn thuộc 1 trong 10 lớp nằm trong thư mục có tên lớp tương ứng.

|  |  |  |
| --- | --- | --- |
| Nhãn | Số phần tử trong tập huấn luyện | Số phần tử trong tập kiểm tra |
| cntt | 150 | 50 |
| giai tri | 150 | 50 |
| giao duc | 150 | 50 |
| kinh doanh | 150 | 50 |
| nau an | 150 | 50 |
| phap luat | 150 | 50 |
| suc khoe | 150 | 50 |
| the gioi | 150 | 50 |
| the thao | 150 | 50 |
| tinh yeu | 150 | 50 |
| Tổng cộng | 1500 | 500 |

Bảng 1: Thông tin về tập dữ liệu VB2000

Đánh giá chung tập dữ liệu rất đồng đều, không có tình trạng mất cân bằng thuận tiện hơn cho việc huấn luyện và kiểm tra.

Nguồn tập dữ liệu VB2000: <https://drive.google.com/drive/folders/1B2Azpg0AVgHzyfEr3cj17GeICwSqoj0o?usp=sharing>

## Các bước xây dựng mô hình

Để tiến hành giải quyết yêu cầu với tập dữ liệu văn bản Tiếng Việt, trước tiên cần nhập những thư viện cần thiết của Python

import contextlib

import datetime

import matplotlib.pyplot as plt

import numpy as np

import os

import seaborn as sns

import sklearn.preprocessing

import tensorflow as tf

import tensorflow.keras as keras

import transformers

### Xây dựng hàm create\_dataset\_from\_dir

def create\_dataset\_from\_dir(dataset\_dir: str,

le: sklearn.preprocessing.LabelEncoder,

tokenizer:transformers.AutoTokenizer)-> tf.data.Dataset:

labels = tf.io.gfile.listdir(dataset\_dir)

print(labels)

input\_ids = list()

attention\_mask = list()

y= list()

for label in labels:

samples = tf.io.gfile.listdir("{}/{}".format(dataset\_dir, label))

print("{}: {}".format(label, len(samples)))

for sample in samples:

with open("{}/{}/{}".format(dataset\_dir, label, sample), 'r', encoding="utf8", errors="ignore") as f:

encoded\_input = tokenizer(f.read(), padding="max\_length",truncation=True, return\_tensors="tf")

input\_ids.append(encoded\_input["input\_ids"][0])

attention\_mask.append(encoded\_input["attention\_mask"][0])

y.append(label.lower())

return tf.data.Dataset.from\_tensor\_slices(((input\_ids, attention\_mask), le.transform(y))).shuffle(10000).batch(128)

* Hàm **create\_dataset\_from\_dir** nhận những tham số
  + Đường dẫn chứa tập dữ liệu (train hoặc test)
  + 1 label encoder để mã hoá nhãn của lớp
  + tokenizer của mô hình PhoBERT để tiến hành token hoá những dữ liệu văn bản được đọc vào
* Sau đó thực hiện:
  + Đọc những file văn bản trong thư mục và nhãn của đoạn văn đó
  + Mã hoá nhãn bằng label encoder và token hoá đoạn văn bằng PhoBERT tokenizer
  + Trả về kết quả là 1 tập dữ liệu mang cấu trúc **tf.data.Dataset** đã được trộn lên và chia ra thành những đoạn 128 phần tử

### Đọc dữ liệu

# const

VB2000\_path = "VB2000"

train\_set = "train-1500"

test\_set = "test-500"

# load PhoBERT from transformers

PhoBERT\_name = "vinai/phobert-base"

PhoBERT\_model = transformers.TFRobertaModel.from\_pretrained(PhoBERT\_name)

PhoBERT\_tokenizer = transformers.AutoTokenizer.from\_pretrained(PhoBERT\_name,

use\_fast=False)

# label encoder

le = sklearn.preprocessing.LabelEncoder()

le.fit([label.lower() for label in tf.io.gfile.listdir("{}/{}".format(VB2000\_path, train\_set))])

# load n preprocess data

train\_data = create\_dataset\_from\_dir("{}/{}".format(VB2000\_path, train\_set), le, PhoBERT\_tokenizer)

test\_data = create\_dataset\_from\_dir("{}/{}".format(VB2000\_path, test\_set) , le, PhoBERT\_tokenizer)

* Thiết lập những hằng số như:
  + Đường dẫn chứa tập dữ liệu VB2000
  + Tên của tập train và test
* Đọc và khởi tạo mô hình PhoBERT nhằm xử lý văn bản
* Tạo 1 label encoder bằng sklearn
* Gọi hàm **create\_dataset\_from\_dir** ở 2 tập train và test để dễ dàng đưa vào huấn luyện mô hình

### Xây dựng mô hình

# build model

input\_ids = keras.layers.Input(shape=(PhoBERT\_tokenizer.model\_max\_length, ), name="input\_ids" , dtype=tf.int32)

attention\_mask = keras.layers.Input(shape=(PhoBERT\_tokenizer.model\_max\_length, ), name="attention\_mask", dtype=tf.int32)

# DO NOT ASK WHY https://github.com/keras-team/keras/issues/14345#issuecomment-937142751

embeddings = PhoBERT\_model.roberta(input\_ids, attention\_mask)[0]

# reduce dim

x = keras.layers.GlobalMaxPool1D()(embeddings)

x = keras.layers.BatchNormalization()(x)

###################

# sửa từ đoạn này #

###################

x = keras.layers.Dense(128, activation="relu")(x)

####################

# ngưng sửa từ đây #

####################

y = keras.layers.Dense(len(le.classes\_), activation="softmax", name="outputs")(x)

model = keras.Model(inputs=[input\_ids, attention\_mask], outputs=y)

# freeze PhoBERT params

model.layers[2].trainable = False

model.summary()

model.compile(optimizer="adam",

loss=keras.losses.SparseCategoricalCrossentropy(),

metrics=['accuracy'])

* Kiến trúc mô hình:
  + Dữ liệu sau khi được Tokenizer: cho ra kết quả gồm 2 mảng. Từ 2 mảng đấy sẽ được đưa vào mô hình PhoBERT để cho ra các vector đặc trưng của đoạn văn bản tương ứng
    - input\_ids: thể hiện token của đoạn văn bản input
    - attention\_mask: thể hiện xem những vị trí cần chú ý trong input
      * 1 là token đó có trong input
      * 0 là padding được thêm vào cho đủ độ dài của mô hình
  + PhoBERT: mô hình PhoBERT được load bằng thư viện transformers cụ thể ở đây sử dụng mô hình **vinai/phobert-base**
    - Cần đánh dấu các thông số của mô hình PhoBERT là không huấn luyện được nhằm tránh tình trạng mô hình PhoBERT bị cập nhật trong lúc huấn luyện
  + Các tầng ẩn: nơi sẽ được đánh giá thông qua nhiều kiến trúc mô hình khác nhau để chọn ra kiến trúc có kết quả dự đoán chính xác nhất
  + Kết quả: tầng kêt quả của mô hình gồm 10 nơ ron với hàm kích hoạt softmax
* Những thông số khác của mô hình:
  + Sử dụng giải thuật tối ưu Adam
  + Loss: sử dụng SparseCategoricalCrossentropy do:
    - Lớp của tập dữ liệu là 1 số cụ thể không phải dạng one hot
    - Mỗi file chứa đoạn văn chỉ thuộc vào 1 lớp duy nhất
  + Phương thức đánh giá: độ chính xác

## Kết quả huấn luyện

Các giá trị cần quan tâm để thực hiện so sánh giữa các mô hình:

* Accuracy: độ chính xác tập train
* Val\_Accracy: độ chính xác tập test
* Loss: độ lỗi của tập train
* Val\_Loss: độ lỗi tập test

### Kết quả mô hình 1

Mô hình thứ nhất gồm: 1 tầng (1 layer) – Số nút: 64

#### Kết quả độ chính xác mô hình

Hình 1: Biểu đồ minh họa độ chính xác của mô hình thứ nhất

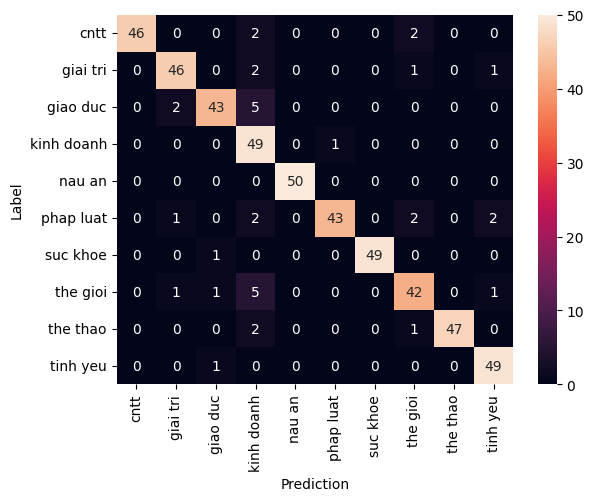
Biểu đồ cho thấy khi càng huấn luyện thì độ chính xác càng tăng.

#### Kết quả độ lỗi mô hình

Hình 2: Biểu đồ minh họa độ lỗi của mô hình thứ nhất

Biểu đồ cho thấy khi càng huấn luyện thì lỗi càng giảm.

#### Mô tả chi tiết phần dự đoán của mô hình



Hình 3: Mô tả chi tiết phần dự đoán của mô hình thứ nhất

Mô hình dự đoán:

* Sai nhiều ở 3 nhãn: Giáo dục (43), Pháp luật (43), Thế giới (42) (<45)
* Độ chính xác cao ở bảy nhãn còn lại (>= 45)

### Kết quả mô hình 2

Mô hình thứ hai gồm: 2 tầng (2 layer) – Số nút 64

#### Kết quả độ chính xác mô hình

Hình 4: Biểu đồ minh họa độ chính xác của mô hình thứ hai

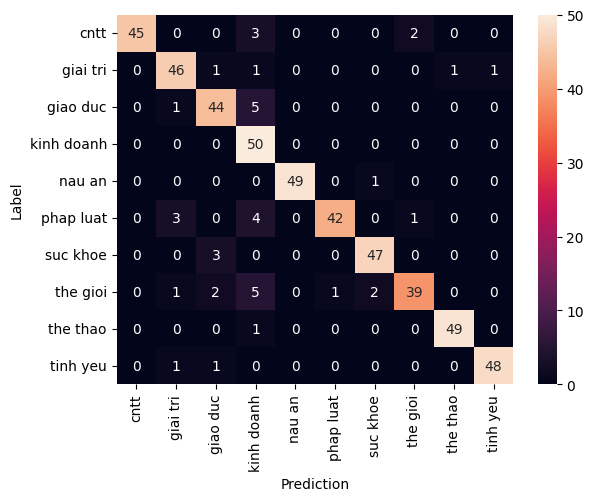
Biểu đồ cho thấy khi càng huấn luyện thì độ chính xác càng tăng.

#### Kết quả độ lỗi mô hình

Hình 5: Biểu đồ minh họa độ lỗi của mô hình thứ hai

Biểu đồ cho thấy khi càng huấn luyện thì lỗi càng giảm.

#### Mô tả chi tiết phần dự đoán của mô hình



Hình 6: Mô tả chi tiết phần dự đoán của mô hình thứ hai

Mô hình dự đoán:

* Sai nhiều 3 nhãn: Giáo dục (44), Pháp luật (42), Thế giới (39) (<45)
* Độ chính xác cao ở bảy nhãn còn lại (>= 45)

### Kết quả mô hình 3

Mô hình thứ 3:

* 3 tầng (3 layer)
* Số nút:
* Tầng một: 512
* Tầng hai: 256
* Tầng ba: 128

#### Kết quả độ chính xác mô hình

Hình 7: Biểu đồ minh họa độ chính xác của mô hình thứ ba

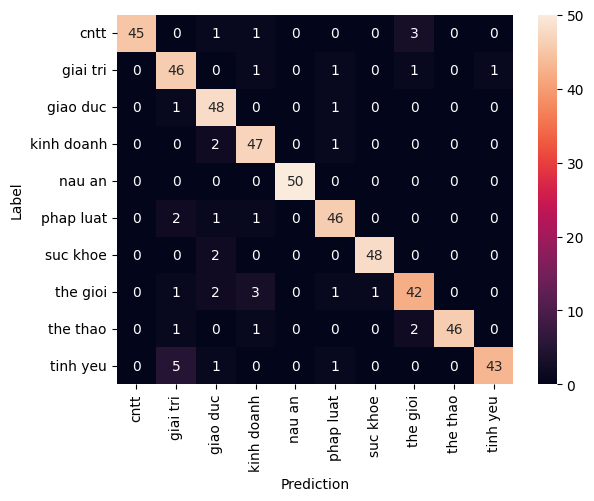
Biểu đồ cho thấy khi càng huấn luyện thì độ chính xác càng tăng.

#### Kết quả độ lỗi mô hình

Hình 8: Biểu đồ minh họa độ lỗi của mô hình thứ ba

Biểu đồ cho thấy khi càng huấn luyện thì lỗi càng giảm.

#### Mô tả chi tiết phần dự đoán của mô hình



Hình 9: Mô tả chi tiết phần dự đoán của mô hình thứ ba

Mô hình dự đoán:

* Sai nhiều ở 2 nhãn: Thế giới (42), Tình yêu (43) (<45)
* Độ chính xác cao ở tám nhãn còn lại (>= 45)

### Kết quả mô hình 4

Mô hình thứ tư: 1 tầng (1 layer) – Số nút: 128

#### Kết quả độ chính xác mô hình

Hình 10: Biểu đồ minh họa độ chính xác của mô hình thứ tư

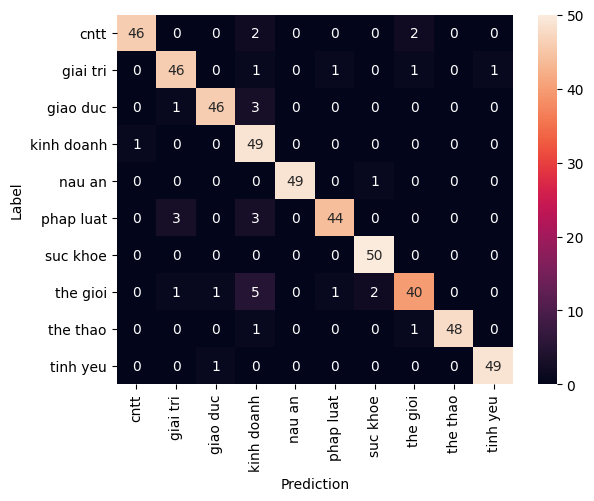
Biểu đồ cho thấy khi càng huấn luyện thì độ chính xác càng tăng.

#### Kết quả độ lỗi mô hình

Hình 11: Biểu đồ minh họa độ lỗi của mô hình thứ ba

Biểu đồ cho thấy khi càng huấn luyện thì lỗi càng giảm.

#### Mô tả chi tiết phần dự đoán của mô hình



Hình 12: Mô tả chi tiết phần dự đoán của mô hình thứ tư

Mô hình dự đoán:

* Sai nhiều ở 2 nhãn: Pháp luật (44), Thế giới (40) (<45)
* Độ chính xác cao ở tám nhãn còn lại (>= 45)

### Kết quả mô hình thứ năm

Mô hình thứ tư: 1 tầng (1 layer) – Số nút: 128 – Không có hàm kích hoạt.

#### Kết quả độ chính xác mô hình

Hình 13: Biểu đồ minh họa độ chính xác của mô hình thứ năm

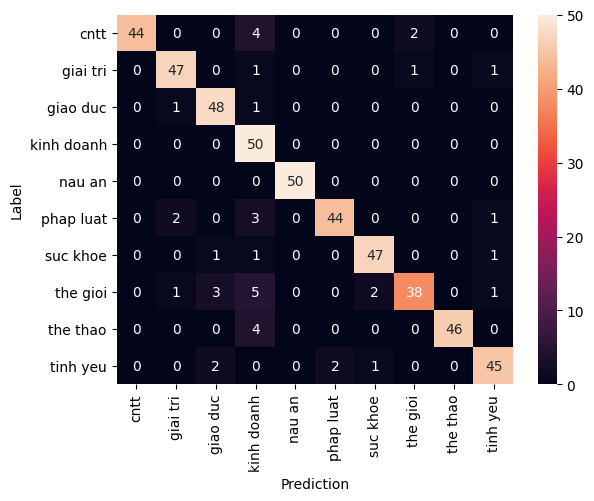
Biểu đồ cho thấy khi càng huấn luyện thì độ chính xác càng tăng.

#### Kết quả độ lỗi mô hình

Hình 14: Biểu đồ minh họa độ lỗi của mô hình thứ năm

Biểu đồ cho thấy khi càng huấn luyện thì lỗi càng giảm.

#### Mô tả chi tiết phần dự đoán của mô hình



Hình 15: Mô tả chi tiết phần dự đoán của mô hình thứ năm

Mô hình dự đoán:

* Sai nhiều ở 3 nhãn: CNTT (44), Pháp luật (44), Thế giới (38) (<45)
* Độ chính xác cao ở bảy nhãn còn lại (>= 45)

### So sánh kết quả

#### So sánh giữa mô hình 1, mô hình 2 và mô hình 3

Ba mô hình có số lượng tầng tăng dần, có hàm kích hoạt. Thực hiện so sánh kết quả huấn luyện ba mô hình trên cùng một tập dữ liệu.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Mô hình 1 | Mô hình 2 | Mô hình 3 |
| Chỉ số val\_accuracy  (lớn nhất) | 0.917999983 | 0.917999983 | 0.924000025 |
| Chỉ số val\_loss  (nhỏ nhất) | 0.297281891 | 0.297281891 | 0.251440376 |
| Số nhãn dự đoán sai nhiều hơn 45 phần tử | 3 | 3 | 2 |
| Số nhãn dự đoán đúng nhiều hơn 45 phần tử | 7 | 7 | 8 |

Bảng 2: So sánh giữa 3 mô hình

**Kết luận:**

Mô hình 3 với 3 tầng (3 layer) đã cho kết quả dự báo chính xác nhất trong 3 mô hình huấn luyện.

#### So sánh giữa mô hình 4 và mô hình 5

Hai mô hình 3 và 4 có cùng 1 tầng, và cùng số nút là 128. Tuy nhiên, mô hình 4 có hàm kích hoạt còn mô hình 5 thì không có hàm kích hoạt. Thực hiện so sánh kết quả huấn luyện của hai mô hình để xem sự khác nhau khi có và không có của hàm kích hoạt trong việc huấn luyện môn hình máy học.

|  |  |  |
| --- | --- | --- |
|  | Mô hình 4 | Mô hình 5 |
| Chỉ số val\_accuracy  (lớn nhất) | 0.934000015 | 0.917999983 |
| Chỉ số val\_loss  (nhỏ nhất) | 0.30119732 | 0.31086567 |
| Số nhãn dự đoán sai nhiều hơn 45 phần tử | 2 | 3 |
| Số nhãn dự đoán đúng nhiều hơn 45 phần tử | 8 | 7 |

Bảng 3: So sánh giữa hai mô hình 3 và mô hình 4

**Kết luận:**

Mô hình 4 với 1 tầng (1 layer) và hàm kích hoạt đã cho kết quả dự báo chính xác nhất trong 2 mô hình huấn luyện. Vì mô hình 4 có hàm kích hoạt điều đó đã làm cho dữ liệu được phân hoạch rõ ràng hơn, nên kết quả huấn luyện mô hình sẽ chính xác hơn mô hình 5 có cùng số tầng, tuy nhiên không có hàm kích hoạt.

#### So sánh giữa mô hình 3 và mô hình 4

Sau khi thực hiện so sánh, hai mô hình có kết quả huấn luyện tốt nhất là mô hình 3 và mô hình 4.

* Mô hình 3 là mô hình gồm 3 tầng (3 layer) và có hàm kích hoạt.
* Mô hình 4 là mô hình gồm 1 tầng (3 layer) và có hàm kích hoạt.

|  |  |  |
| --- | --- | --- |
|  | Mô hình 3 | Mô hình 4 |
| Chỉ số val\_accuracy  (lớn nhất) | 0.924000025 | 0.934000015 |
| Chỉ số val\_loss  (nhỏ nhất) | 0.251440376 | 0.30119732 |
| Số nhãn dự đoán sai nhiều hơn 45 phần tử | 2 | 2 |
| Số nhãn dự đoán đúng nhiều hơn 45 phần tử | 8 | 8 |

Bảng 4: So sánh giữa hai mô hình có kết quả huấn luyện tốt nhất

**Kết luận:**

Cả hai mô hình đều cho kết quả dự đoán về:

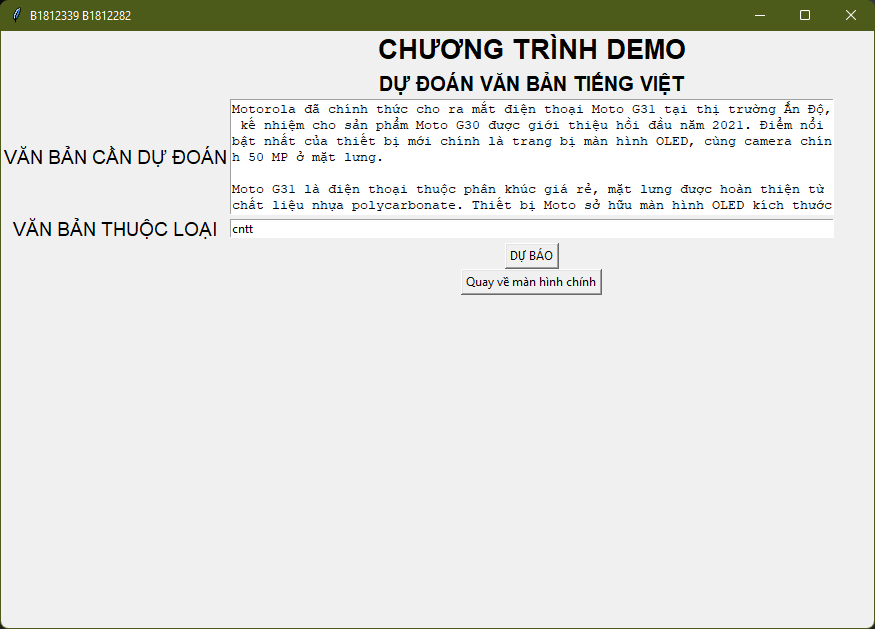
* Số nhãn dự đoán sai nhỏ hơn 45 phần tử: 2 nhãn
* Số nhãn dự đoán đúng nhiều hơn 45 phần tử: 8 nhãn

Với chỉ số val\_accuracy: thì mô hình 4 cho kết quả dự đoán chính xác hơn. Tuy nhiên kết quả dự đoán các phần tử lỗi thì mô hình 3 lại cho kết quả dự đoán chính xác hơn. Vì thế, nghiên cứu này sẽ sử dụng mô hình 4 bao gồm: 3 tầng (3 layer) có các số nút lần lượt là:

* Tầng 1: 512 nút
* Tầng 2: 256 nút
* Tầng 3: 128 nút
* Để thực hiện giải quyết yêu cầu bài toán

## Xây dựng ứng dụng

Sau khi chọn ra được mô hình có kết quả tốt nhất, có thể xây dựng 1 ứng dụng đơn giản đọc vào 1 đoạn văn bản và xuất ra kết quả là nhãn của đoạn văn bản đó



Hình 16: Giao diện ứng dụng

Sơ lược các bước thực thi của ứng dụng:

* Dựng GUI với các nút, ô cần thiết
* Load mô hình tốt nhất, PhoBERT tokenizer và mảng nhãn văn bản
* Khi người dùng ấn dự báo:
  + Lấy dữ liệu text từ phía người dùng nhập vào
  + Đưa dữ liệu đó qua bộ tokenizer
  + Đưa kết quả bộ tokenizer vào mô hình
  + Lấy kết quả của mô hình ở vị trí có khả năng cao nhất bằng argmax()
  + Hiển thị nhãn tương ứng lên GUI

# KẾT QUẢ ĐẠT ĐƯỢC

Nghiên cứu xây dựng 5 mô hình bao gồm:

* Mô hình 1: 1 tầng – số nút 64.
* Mô hình 2: 2 tầng – số nút 64.
* Mô hình 3: 3 tầng – số nút lần lượt là 512/256/128.
* Mô hình 4: 1 tầng – số nút 128.
* Mô hình 5: 1 tầng – số nút 128 (không có hàm kích hoạt).

Tập dữ liệu được sử dụng để huấn luyện mô hình là: Dữ liệu văn bản Tiếng Việt có hai tập dữ liệu lần lượt là test (500), train (1500). Sau khi huấn luyện qua năm dạng mô hình khác nhau, nghiên cứu quyết định lựa chọn mô hình thứ 3 là mô hình có kết quả huấn luyện tốt nhất.

# TÀI LIỆU THAM KHẢO

<https://huggingface.co/vinai/phobert-base>

<https://github.com/VinAIResearch/PhoBERT>

<https://github.com/keras-team/keras/issues/14345#issuecomment-937142751>

# PHỤ LỤC