BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

BỘ MÔN CƠ ĐIỆN TỬ

___oOo___

BÀI BÁO

MÔN HỌC TRÍ TUỆ NHÂN TẠO

GVHD**: PGS.TS Nguyễn Trường Thịnh**

SVTH**: Trần Triệu Vĩ** MSSV**: 19146301**

Thành phố Hồ Chí Minh, ngày 20 tháng 6 năm 2022

1. Giới thiệu bài toán

Bài toán phân tích cảm xúc đánh giá của khách hàng về sản phẩm điện thoại thông minh

- Đầu vào bài toán (Input): Tập dữ liệu về các bình luận đáng giá của khách hàng về hoat động mua bán và chất lượng sản phẩm điện thoại thông minh

Hình Đánh giá của khách hàng về điện thoại thông minh trên trang web: https://www.thegioididong.com/

- Đầu ra bài toán (Output): Cảm xúc của bình luận được dự đoán bằng mô hình

Ví dụ:

Đánh giá	Cảm xúc
"Máy có thiết kế đẹp, pin trâu, loa to rõ"	Tích cực
"Máy bắt wifi kém, máy mau nóng và pin mau hết"	Tiêu cực

2. Quá trình thực hiện

2.1 Thu thập và phân tích dữ liệu

2.1.1 Thu thập dữ liệu

Tập dữ liệu được sử dụng để huấn luyện mô hình trong đề tài này là tập dữ liệu UIT-ViSFD - Vietnamese Smartphone Feedback Dataset của nhóm tác giả với đại diện là tác giả Luong Luc Phan. Tập dữ liệu bao gồm các đánh giá của khách hàng về điện thoại thông minh được trích xuất từ website thương mại điện tử lớn ở Việt Nam. Các thuộc tính trong bộ dữ liệu:

- comment: Nội dung đánh giá của khách hàng
- n_star: Số sao người mua, người dùng đánh giá cho điện thoại thông minh

- data_time: Thời gian đánh giá được đăng tải
- label: các nhãn của bình luận

Trong bài toán này chỉ sử dụng 10010 điểm dữ liệu để chia thành 2 tập dữ liệu:

- Dữ liệu huấn luyện (Training Data): [0:9000] tức 9000 dữ liệu đầu tiên
- Dữ liệu đánh giá (Validation Data): [9000:10010] tức 1010 dữ liệu cuối cùng.

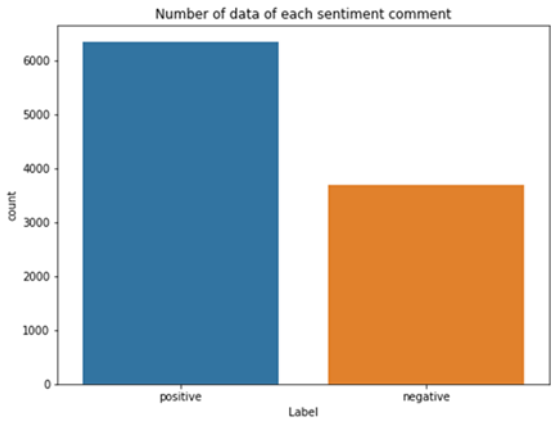
Gán nhãn và lấy những thuộc tính cần thiết của dữ liệu:

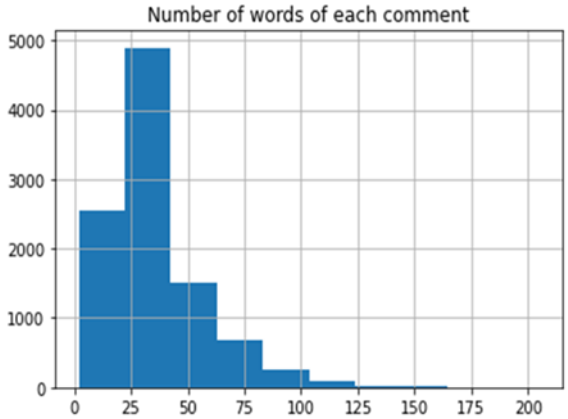
Xây dựng hàm create_label ở trên sẽ gắn nhãn cho dữ liệu: nếu số sao của đánh giá đó < 4 sao thì đó là đánh giá tiêu cực và >= 4 sao là đánh giá tích cực.

```
# Data labeling function
def create_label(main_data):
  conditions = [
    (main_data['n_star'] <= 3),
    (main_data['n_star'] >=4),
    ]
  values = [0, 1]
  main_data['label'] = np.select(conditions, values)
  main_data =  main_data[['index','comment','label']]
  return main_data
```

2.1.2 Phân tích dữ liệu

Dữ liệu bao gồm 6333 đánh giá tích cực và 3677 đánh giá tiêu cực. Độ dài của các đánh giá tập trung vào khoảng 100 từ trở lại.





Số lượng đánh giá tiêu cực, tích cực *Phân bố số lượng từ trong mỗi đánh giá*

2.2 Tiền xử lý dữ liệu

- **Xóa các giá trị bị thiếu, các giá trị đặc biệt, các chữ số**

Xây dựng hàm preprocess_data để xóa các giá trị na/nun, hàm preprocessing_text để loại bỏ các kí tự đặc biệt và chữ số

```
# Data preprocessing function
def preprocess_data(data):
  # Renove Na/Nul values
  data = data.dropna(axis = 0,subset=['comment'])
  return data
```

```
# Comment preprocessor function in data
def preprocessing_text(text):
  # Remove special character
  text = text.str.replace('[^\w\s]','')
  # Remove digit
  text = text.replace('\d', '', regex=True)
  # Standardize acronyms for the word: 'khong'
  text = [process_special_word(i) for i in text]
  return text
```

- **Xử lý từ mang nghĩa "không"**

Xây dựng hàm process_special_word để ghép từ mang nghĩa phủ định gần giống như từ "không" với từ kế tiếp nó trong câu thông qua file từ điển not.txt được chuẩn bị sẵn. Các từ có trong từ điển à: không, k, ko, vô, chẳng, đếch, chưa, đéo, kém, nỏ, not, chả, kh, hong, hem, hum, ch, chưa_được, ch_dc, chua_dc

```
[11] # Load file containing acronyms for the word: 'khong'
     file = open('not.txt', 'r', encoding="utf8")
     not_lst = file.read().split('\n')
     file.close()
```

```
[12] # Standardize acronyms for the word: 'khong'
     def process_special_word(text):
         for khong in not_lst:
           new_text = ''
           text_lst = text.split()
           i= 0
           if khong in text_lst:
               while i <= len(text_lst) - 1:
                   word = text_lst[i]
                   if  word == khong:
                       next_idx = i+1
                       if next_idx <= len(text_lst) -1:
                           word = word +'_'+ text_lst[next_idx]
                       i= next_idx + 1
                   else:
                       i = i+1
                   new_text = new_text + word + ' '
           else:
               new_text = text
           text=new_text
         return new_text.strip()
```

Kết quả đạt được của quá trình tiền xử lý dữ liệu:

```
feedback = 'Điện thoại mới mua hôm 20/6 mau hết pin mà hum được bảo hành!!! 😢'
print("Trước xử lý: \n\t", feedback)
print("Sau xử lý: \n\t", preprocess(feedback))
```

```
Trước xử lý:
        Điện thoại mới mua hôm 20/6 mau hết pin mà hum được bảo hành!!! 😢
Sau xử lý:
        điện_thoại mới mua hôm mau hết pin mà hum_được bảo_hành
```

2.3 Vector hóa dữ liệu

- **Xây dựng mô hình Word2vec:** biến đổi bộ từ vựng từ chính các đánh giá của khách hàng trong tập dữ liệu thành không gian vector

```
model.wv.most_similar("xấu")
```

```
[('không_đẹp', 0.9317137598991394),
 ('không_nét', 0.9271178841590881),
 ('zoom', 0.9192898273468018),
 ('ảo', 0.9132571816444397),
 ('nhạt', 0.9109840393066406),
 ('vỡ', 0.9101008176803589),
 ('selfie', 0.9060805439949036),
 ('bệt', 0.8993314504623413),
 ('ko_đẹp', 0.8925150632858276),
 ('k_đẹp', 0.8905947208404541)]
```

- **Vector hóa đánh giá:** đánh giá của khách hàng sẽ được vector hóa sang không gian 128 chiều và do chỉ lấy 100 từ đầu tiên của đánh giá để xử lý nên mỗi đánh giá sau khi vector hóa sẽ có kích thước 100x128. Hàm vector hóa đánh giá có cấu trúc như bên dưới:

```
# Embedding word into matrix function
def comment_embedding(comment):
    matrix = np.zeros((max_seq, embedding_size))
    words = comment.split()
    lencmt = len(words)
    n_cau = 0
    for i in range(max_seq):
        indexword = i % lencmt
        n_cau = i//lencmt
        if (max_seq - n_cau*lencmt < lencmt):
            break
        if(words[indexword] in word_labels):
            matrix[i] = model_embedding[words[indexword]]
    matrix = np.array(matrix)
    return matrix
```

Kết quả đạt được của quá trình vector hóa dữ liệu:

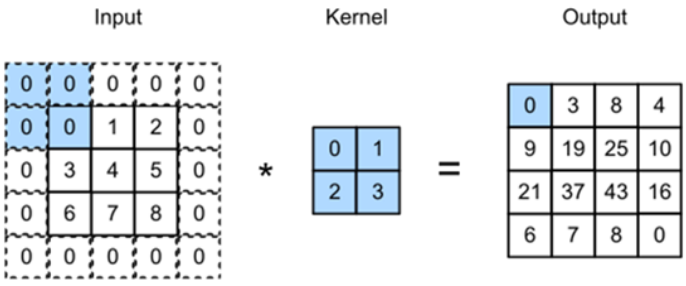
```
feedback = 'Đã chuyển từ XS Mã qua, do ko phải là OLED nên cũng rất ít tốn pin.'
print(comment_embedding(feedback))
print('Kích thước của vector đánh giá:\t',comment_embedding(feedback).shape)
```

```
[[ 0.          0.          0.         ...  0.          0.
   0.        ]
 [ 0.31778145 -0.25731125 -0.47602418 ... -0.23017888 -0.09705949
  -0.00088737]
 [-0.01561598 -0.08790187 -0.54559284 ... -0.24832276  0.01362987
   0.11153618]
 ...
 [ 0.          0.          0.         ...  0.          0.
   0.        ]
 [ 0.          0.          0.         ...  0.          0.
   0.        ]
 [ 0.          0.          0.         ...  0.          0.
   0.        ]]
Kích thước của vector đánh giá:  (100, 128)
```

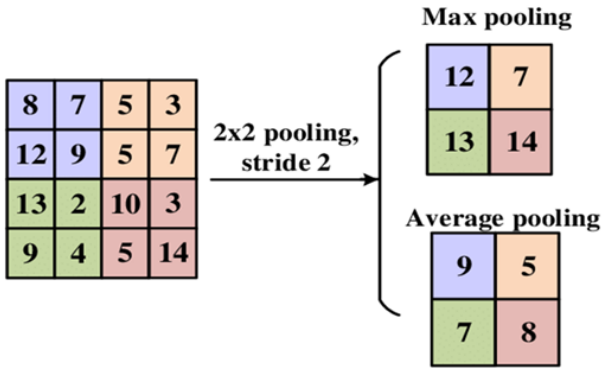
2.4 Xây dựng và huấn luyện mô hình

Mô hình được xây dựng trong thuật toán này là mô hình Convolutional Neural Network (CNN) được hiểu là một mạng thần kinh tích chập. Một mô hình CNN bao gồm các lớp quang trọng:

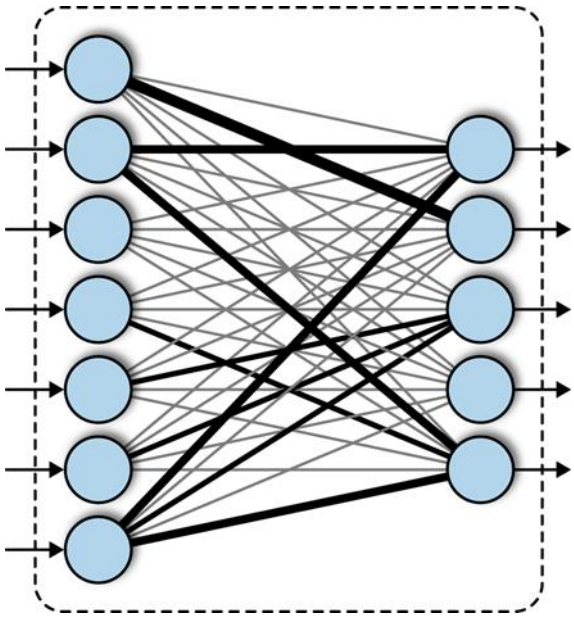
- **Convolutional Layer:** Là một hidden layer, gồm một tập các feature maps là các bản scan từ input đầu vào ban đầu. Convolutional Filter hay còn gọi là Kernel, là một ma trận sẽ quét ma trận dữ liệu đầu vào từ trái sang phải, từ trên xuống dưới.



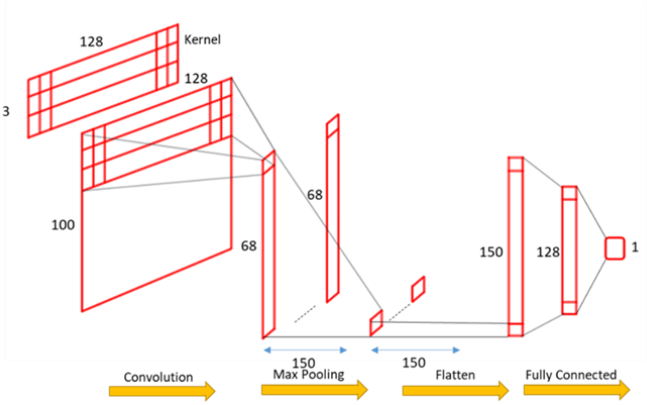
- **Pooling Layer:** được biết đến với hai loại phổ biến: Max Pooling và Average Pooling. Các Pooling window sẽ trượt trên ma trận dữ liệu đầu vào, sau đó chọn 1 giá trị lớn nhất đối với Max Pooling hay lấy giá trị trung bình làm đại diện đối với Average Pooling



- **Fully Connected Layer:** là layer mà mối nút của nó kết nối với tất cả các nút ở layer kế trước nó.



Chi tiết mô hình CNN được xây dựng:

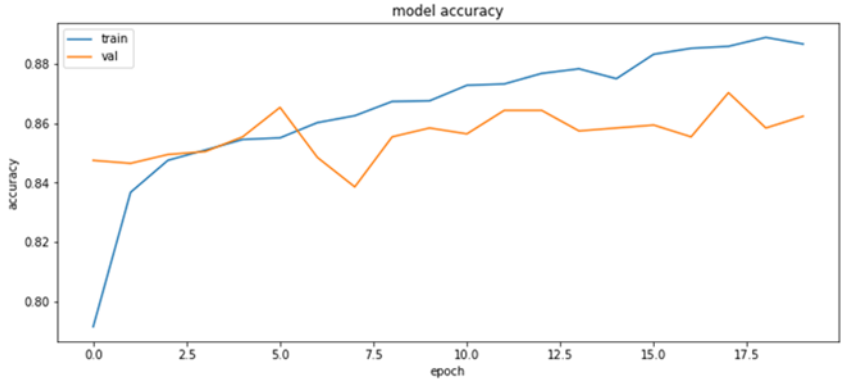


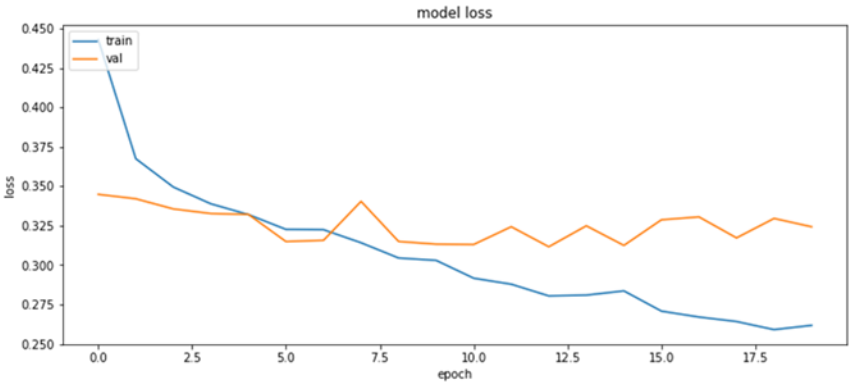
Tóm tắt mô hình CNN:

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_5 (Conv2D)           (None, 98, 1, 150)        57750

 max_pooling2d_5 (MaxPooling  (None, 1, 1, 150)        0
 2D)

 dropout_5 (Dropout)         (None, 1, 1, 150)         0

 flatten_5 (Flatten)         (None, 150)               0

 dense_10 (Dense)            (None, 128)               19328

 dense_11 (Dense)            (None, 1)                 129

=================================================================
Total params: 77,207
Trainable params: 77,207
Non-trainable params: 0
```

3. Kết quả thực nghiệm

Mô hình đạt độ chính xác trên Training data là 88,67% và trên Validation data là 86,24 %.





Dự đoán một số đánh giá dễ bị phân tích sai:

```
predict('máy này chơi game không được')
```

```
Tiêu cực
0.34
```

```
predict('máy bắt wifi chưa được tốt cho lắm')
```

```
Tiêu cực
0.29
```

Giao diện website dự đoán đánh giá khách hàng:

TÀI LIỆU THAM KHẢO

1. Pham Dinh Khanh, Bài 3 - Mô hình Word2Vec, https://phamdinhkhanh.github.io/2019/04/29/ModelWord2Vec.html
2. Nguyễn Văn Hiếu, Flask python là gì? Thư viện flask trong lập trình Python, https://nguyenvanhieu.vn/thu-vien-flask-python-la-gi/
3. Got It Vietnam, Tổng quan thư viện NumPy trong Python, https://vn.got-it.ai/blog/tong-quan-thu-vien-numpy-trong-python
4. Nguyen Van Hoang, Giới thiệu về Pandas (một thư viện phổ biến của Python cho việc phân tích dữ liệu), https://viblo.asia/p/gioi-thieu-ve-pandas-mot-thu-vien-pho-bien-cua-python-cho-viec-phan-tich-du-lieu-aWj53Nnel6m
5. Nguyễn Văn Hiếu, [Khóa học tensorflow] Bài 1 – Tổng quan về thư viện Tensorflow, https://nguyenvanhieu.vn/thu-vien-tensorflow/
6. Luong Luc Phan, UIT-ViSFD, https://github.com/LuongPhan/UIT-ViSFD
7. Nguyễn Hồng Đoàn, Sentiment-Analysis-VLSP2016-BaseML, https://github.com/doanbk/Sentiment-Analysis-VLSP2016-BaseML
8. Trà My, Convolutional Neural Network là gì? Cách chọn tham số cho Convolutional Neural Network chuẩn chỉnh, https://wiki.tino.org/convolutional-neural-network-la-gi/