ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

CÁC KỸ THUẬT HỌC SÂU VÀ ỨNG DỤNG

ĐỒ ÁN CUỐI KỲ: TEXT CLASSIFICATION VỚI DNN VÀ TF-IDF

GIẢNG VIÊN: NGUYỄN VINH TIỆP

NHÓM THỰC HIỆN:

- Lê Nguyễn Minh Huy – 20521394

LỚP: CS431.N11

Tp. Hồ Chí Minh, Ngày 20 tháng 11 năm 2022

MỤC LỤC

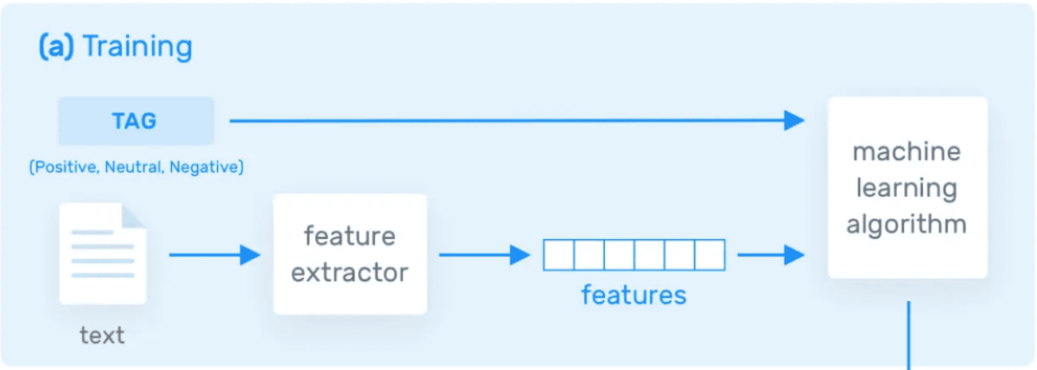
1. **Các vấn đề đã giải quyết được**

   - Xây dựng được Protocol đánh giá chung
   - Xây dựng được các Hyperparameters cần tuning
   - Xây dựng được phương pháp tuning các Hyperparameters
   - Xây dựng được mạng DNN phân loại các tin tức tiếng Việt
   - So sánh mô hình của bản thân với các mô hình khác trong cùng bài toán
   - Tối ưu hóa được bộ Hyperparameters, tăng hiệu suất của mô hình có sẵn.

2. **Giới thiệu bài toán**

   - *Phân loại văn bản (Text Classification)* là bài toán thuộc nhóm học có giám sát (Supervised learning) trong học máy. Bài toán này yêu cầu dữ liệu cần có nhãn (label). Mô hình sẽ học từ dữ liệu có nhãn đó, sau đó được dùng để dự đoán nhãn cho các dữ liệu mới mà mô hình chưa gặp.
   - Lấy ví dụ, bạn cần xây dựng một mô hình học máy để dự đoán chủ đề (Kinh tế, Xã hội, Thể thao,…) của một bài báo bất kỳ. Khi đó, bạn cần rất nhiều dữ liệu có gán nhãn; tức là bạn cần rất nhiều bài báo mà mỗi bài báo đó chúng ta phải biết trước nó nằm trong chủ đề nào rồi. Vấn đề dữ liệu có lẽ là vấn đề nan giải nhất của mô hình học giám sát này
   - Mục tiêu của một hệ thống phân loại văn bản là nó có thể tự động phân loại một văn bản cho trước, để xác định xem văn bản đó thuộc thể loại nào. Một số ứng dụng của hệ thống phân loại như:

     - Hiểu được ý nghĩa, đánh giá, bình luận của người dùng từ mạng xã hội.
     - Phân loại email là spam hay không spam.
     - Tự động gắn thẻ cho những truy vấn, tìm kiếm của người dùng.
     - Phân loại các bài báo điện tử.



- Giai đoạn (a): Huấn luyện (training) là giai đoạn học tập của mô hình phân loại văn bản. Ở bước này, mô hình sẽ học từ dữ liệu có nhãn (trong ảnh trên nhãn là Positive, Negative, Neutral). Dữ liệu văn bản sẽ được số hóa thông qua bộ trích xuất đặc trưng (feature extractor) để mỗi mẫu dữ liệu trong tập huấn luyện trở thành 1 vector nhiều chiều (đặc trưng). Thuật toán máy học sẽ học và tối ưu các tham số để đạt được kết quả tốt trên tập dữ liệu này. Nhãn của dữ liệu được dùng để đánh giá việc mô hình học tốt không và dựa vào đó để tối ưu.
- Giai đoạn (b): Dự đoán (prediction), là giai đoạn sử dụng mô hình học máy sau khi nó đã học xong. Ở giai đoạn này, dữ liệu cần dự đoán cũng vẫn thực hiện các bước trích xuất đặc trưng. Mô hình đã học sau đó nhận đầu vào là đặc trưng đó và đưa ra kết quả dự đoán.

3. Tổng quan đề tài

- Bài toán phân loại văn bản là một bài toán học giám sát (supervised learning) trong học máy (machine learning), bởi vì nội dung của văn bản đã được gán nhãn, và được sử dụng để thực hiện phân loại. Để giải quyết một bài toán phân loại văn bản, ta thực hiện 4 bước:
  i. Chuẩn bị dữ liệu (Dataset Preparation)
  ii. Xử lý thuộc tính của dữ liệu (Feature Engineering)
  iii. Xây dựng mô hình (Build Model)
  iv. Tinh chỉnh mô hình và cải thiện hiệu năng (Improve Performance)
- Cụ thể, ta sẽ phân loại bài báo tiếng Việt vào để xác định bài báo đó thuộc thể loại nào trong các thể loại: **Chính trị xã hội, Đời sống, Khoa học, Kinh doanh, Pháp luật, Sức khỏe, Thế giới, Thể thao, Văn hóa, Vi tính.**
- Bên cạnh đó, chúng ta sẽ thử phân loại với nhiều thể loại hơn với 27 thể loại: **Âm nhạc, Ẩm thực, Bất động sản, Bóng đá, Chứng khoán, Cúm gà, Cuộc sống đó đây, Du học, Du lịch, Đường vào WTO, Gia đình, Giải trí tin học, Giáo dục, Giới tính, Hacker & Virus, Hình sự, Không gian sống, Kinh doanh quốc tế, Làm đẹp, Lối sống, Mua sắm, Mỹ thuật, Sân khấu điện ảnh, Sản phẩm tin học mới, Tennis, Thế giới trẻ, Thời trang.**

4. Dataset

- Chúng em sử dụng dataset tiếng Việt được cung cấp bởi các tác giả: *Cong Duy Vu Hoang, Dien Dinh, Le Nguyen Nguyen và Quoc*

Hung Ngo trong hội nghị *IEEE International Conference on Research, Innovation and Vision for the Future (RIVF 2007)* với bài báo là *A Comparative Study on Vietnamese Text Classification Methods.*

- Bộ dataset được thu thập từ các trang báo Việt Nam, như là VNExpress, Tuổi trẻ, Thanh niên, và Người lao động.
- Thông tin chi tiết về bộ dataset: A Large-scale Vietnamese News Text Classification Corpus
- Nhóm sẽ thực nghiệm trên cả 2 tập là **VNTC10** và **VNTC27**, tức là bộ dataset VNTC đối với cả hai trường hợp là 10 chủ đề và 27 chủ đề.
- Tập data **VNTC10** bao gồm 10 chủ đề: Chính trị xã hội, Đời sống, Khoa học, Kinh doanh, Pháp luật, Sức khỏe, Thế giới, Thể thao, Văn hóa, Vi tính. Trong đó, bao gồm 84132 tin tức.
- Bộ dataset **VNTC27** bao gồm 27 chủ đề tất cả, bao gồm: Âm nhạc, Ẩm thực, Bất động sản, Bóng đá, Chứng khoán, Cúm gà, Cuộc sống đó đây, Du học, Du lịch, Đường vào WTO, Gia đình, Giải trí tin học, Giáo dục, Giới tính, Hacker & Virus, Hình sự, Không gian sống, Kinh doanh quốc tế, Làm đẹp, Lối sống, Mua sắm, Mỹ thuật, Sân khấu điện ảnh, Sản phẩm tin học mới, Tennis, Thế giới trẻ, Thời trang. Trong đó, có 26451 tin tức tất cả.

5. Xử lý dữ liệu

- Chúng ta hãy cùng đọc thử một bài báo mẫu:
  Thủ tướng Đức nhận lời tham dự lễ kỷ niệm D-Day Thủ tướng Gerhard Schroeder sẽ trở thành nguyên thủ Đức đầu tiên tham dự lễ kỷ niệm ngày quân đồng minh đổ bộ lên bãi biển Normandy trong Thế chiến II (mang mật danh D-Day) vào tháng 6 tới. Ông đã chấp nhận lời mời tham gia lễ kỷ niệm 60 năm ngày D-Day của Tổng thống Pháp Jacque Chirac. Phát ngôn viên của Berlin cho biết: "Tổng thống Chirac đã mời Thủ tướng Schroeder từ trước lễ Giáng sinh và ông đã nhận lời ngay. Thủ tướng cảm thấy rất vui khi được mời". Năm 1994, cố

tổng thống Pháp Francois Mitterrand đã không mời cựu thủ tướng Đức Helmut Kohl đến dự lễ kỷ niệm 50 năm sự kiện D-Day. Giới chức Pháp tuyên bố rằng, việc mời thủ tướng Schroeder tham dự lễ kỷ niệm 60 năm sự kiện D-Day là một hành động mang tính biểu tượng nhằm củng cố bầu không khí hòa bình lâu dài giữa hai nước. Pháp và Đức hy vọng từ đây họ có thể chôn vùi những hận thù quá khứ từng đẩy hai nước lâm vào 2 cuộc đại chiến trong thế kỷ trước. Đúng 1h30' sáng 6/6/1944, Mỹ và Anh bất ngờ cho quân đổ bộ lên bãi biển Normandy để giải phóng Pháp khỏi sự chiếm đóng của phát xít Đức, mở mặt trận thứ hai ở Tây Âu. Cuộc hành quân lịch sử vào giai đoạn cuối của Thế chiến II này đã góp phần cùng Hồng quân Liên Xô, đang tổng tiến công quân Đức ở mặt trận Đông Âu, tiêu diệt đế chế của Hitler. Hàng năm cứ đến ngày 6/6, các cựu chiến binh và chính trị gia đến từ những nước có binh sĩ tham gia cuộc đổ bộ năm 1944 lại tập trung tại bãi biển Normandy để làm lễ kỷ niệm. Khi diễn ra sự kiện D-Day, Thủ tướng Gerhard Schroeder mới được 2 tháng tuổi (ông sinh ngày 7/4/1944). Nhà lãnh đạo Đức không bao giờ biết mặt cha mình vì ông đã bỏ mạng lúc đang tham chiến ở Romania không lâu sau khi Schroeder chào đời.

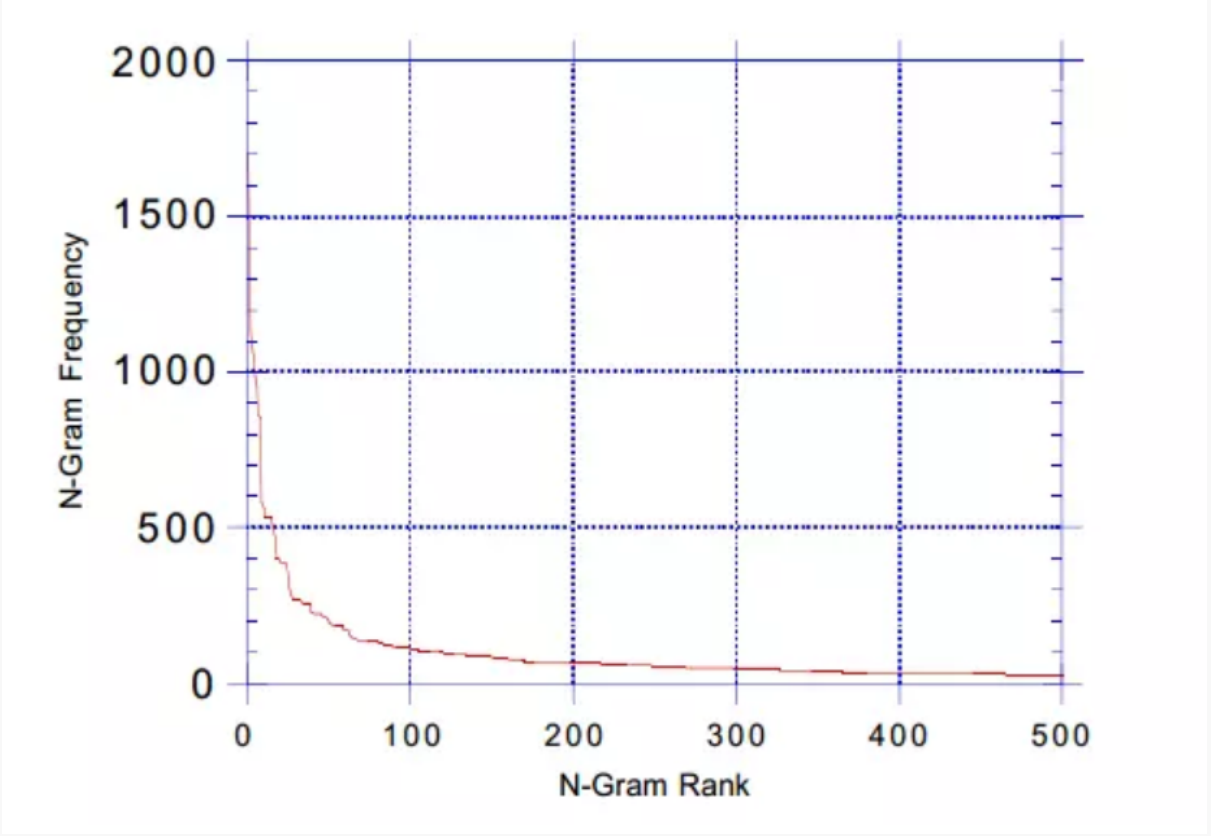
- Đây là bài báo thuộc thể loại *Thế giới*. Điều mà chúng ta thắc mắc ở đây là, từ một văn bản như thế này, làm sao để máy tính có thể hiểu được văn bản để phân loại nó? Câu trả lời là: Máy tính chỉ có thể hiểu được dữ liệu ở dạng số, vì vậy chúng ta cần phải chuyển dữ liệu ở dạng ký tự về dữ liệu dạng số, sau đó chúng ta sẽ sử dụng dữ liệu dạng số này để huấn luyện cho máy tính phân loại các văn bản.
- Trước hết, chúng ta cần phải loại bỏ những ký tự đặc biệt trong văn bản ban đầu như dấu chấm, dấu phẩy, dấu mở đóng ngoặc,... bằng cách sử dụng thư viện *gensim*. Sau đó chúng ta sẽ sử dụng thư viện *PyVi* để tách từ tiếng Việt. Một điểm đặc biệt trong văn bản tiếng Việt đó là một từ có thể được kết hợp bởi nhiều tiếng khác nhau, ví dụ như: sử_dụng, bắt_đầu,... khác với tiếng Anh và một số ngôn ngữ khác, các từ được phân cách nhau bằng khoảng trắng: use some examples, i love you... Vì vậy chúng ta cần tách từ để có thể đảm bảo ý nghĩa của từ được toàn vẹn.
- Để xử lý các ký tự đặc biệt, ta sử dụng hàm *simple_preprocess()*

- có sẵn trong thư viện *gensim*. Để tách các từ trong văn bản tiếng Việt, ta sử dụng thư viện *Vitokenize*
- Sau khi loại bỏ được các ký tự đặc biệt và gom nhóm thành các từ liên quan toàn vẹn về ý nghĩa, chúng ta sẽ chia mỗi bài báo thành dạng các cặp (x,y). Trong đó x là nội dung bài báo, y là nhãn của bài báo đó.
- Kết quả thu được có dạng như sau:
  thủ_tướng đức nhận_lời tham_dự lễ kỷ_niệm day thủ_tướng gerhard schroeder sẽ trở_thành nguyên_thủ đức đầu_tiên tham_dự lễ kỷ_niệm ngày quân đồng_minh đổ_bộ lên bãi biển normandy trong thế_chiến ii mang mật_danh day vào tháng tới ông đã chấp_nhận lời mời tham_gia lễ kỷ_niệm năm ngày day của tổng_thống pháp jacque chirac phát_ngôn_viên của berlin cho biết tổng_thống chirac đã mời thủ_tướng schroeder từ trước lễ giáng_sinh và ông đã nhận_lời ngay thủ_tướng cảm_thấy rất vui khi được mời năm cố tổng_thống pháp francois mitterrand đã không mời cựu_thủ_tướng đức helmut kohl đến dự lễ kỷ_niệm năm sự_kiện day giới_chức pháp tuyên_bố rằng việc mời thủ_tướng schroeder tham_dự lễ kỷ_niệm năm sự_kiện day là một hành_động mang tính biểu_tượng nhằm củng_cố bầu không_khí hòa_bình lâu_dài giữa hai nước pháp và đức hy_vọng từ đây họ có_thể chôn vùi những hận_thù quá_khứ từng đẩy hai nước lâm vào_cuộc đại_chiến trong thế_kỷ trước đúng sáng mỹ và anh bất_ngờ cho quân đổ_bộ lên bãi biển normandy để giải_phóng pháp khỏi sự chiếm_đóng của phát_xít đức mở_mặt_trận thứ hai tây âu cuộc hành_quân lịch_sử vào giai_đoạn cuối của thế_chiến ii này đã góp_phần cùng hồng_quân liên xô đang tổng tiến_công quân đức mặt_trận đông âu tiêu_diệt đế_chế của hitler hàng năm cứ đến ngày các cựu_chiến_binh và chính_trị_gia đến từ những nước có binh_sĩ tham_gia cuộc đổ_bộ năm lại tập_trung tại bãi biển normandy để làm lễ kỷ_niệm khi diễn ra sự_kiện day thủ_tướng gerhard schroeder mới được tháng tuổi ông sinh ngày nhà lãnh_đạo_đức không bao_giờ biết mặt cha mình vì ông đã bỏ_mạng lúc đang tham_chiến romania không lâu sau khi schroeder chào_đời
- Sau đó chúng ta sẽ lưu dữ liệu đã xử lý lại với thư viện *pickle* để tiện cho các lần sử dụng.
- Chúng ta làm tương tự đối với dữ liệu test.

6. Mô hình

- **Word embedding**
  - **TF - IDF**
    - Viết tắt của thuật ngữ tiếng Anh Term *Frequency – Inverse Document Frequency*, *TF-IDF* là trọng số của một từ trong văn bản thu được qua thống kê thể hiện mức độ quan trọng của từ này trong một văn bản, mà bản thân văn bản đang xét nằm trong một tập hợp các văn bản.
    - Thuật toán này thường được sử dụng vì: trong ngôn ngữ luôn có những từ xảy ra thường xuyên với các từ khác. Và một trong những phát biểu nổi tiếng nhất Zipf's law phát biểu về vấn đề này như sau:

      The nth most common word in a human language text occurs with a frequency inversely proportional to n.



    - Có nghĩa là luôn có một tập các từ mà tần số xuất hiện, sử dụng nhiều hơn các từ khác , điều này đúng trong bất kỳ ngôn ngữ nào. Chính vì vậy ta cần có một phương pháp để làm mịn đường cong tần số

trên hay là việc cân bằng mức độ quan trọng giữa các từ

- **Cách tính trọng số TF - IDF**
  - **TF: Term Frequency** : dùng để ước lượng tần suất xuất hiện của từ trong văn bản. Tuy nhiên với mỗi văn bản thì có độ dài khác nhau, vì thế số lần xuất hiện của từ có thể nhiều hơn . Vì vậy số lần xuất hiện của từ sẽ được chia độ dài của văn bản (tổng số từ trong văn bản đó)

$$TF(t, d) = \frac{Số\ văn\ bản\ có\ chứa\ từ\ t}{Tổng\ số\ văn\ bản\ trong\ tập\ mẫu\ D}$$

  - ***IDF:*** **Inverse Document Frequency**: dùng để ước lượng mức độ quan trọng của từ đó như thế nào . Khi tính tần số xuất hiện tf thì các từ đều được coi là quan trọng như nhau. Tuy nhiên có một số từ thường được được sử dụng nhiều nhưng không quan trọng để thể hiện ý nghĩa của đoạn văn , ví dụ :
- Từ nối: và, nhưng, tuy nhiên, vì thế, vì vậy, ...
- Giới từ: ở, trong, trên, ...
- Từ chỉ định: ấy, đó, nhỉ, ...

Vì vậy ta cần giảm đi mức độ quan trọng của những từ đó bằng cách sử dụng IDF :

$$IDF(t, D) = log\ \frac{Tổng\ số\ văn\ bản\ trong\ tập\ mẫu\ D}{Số\ văn\ bản\ có\ chứa\ từ\ t}$$

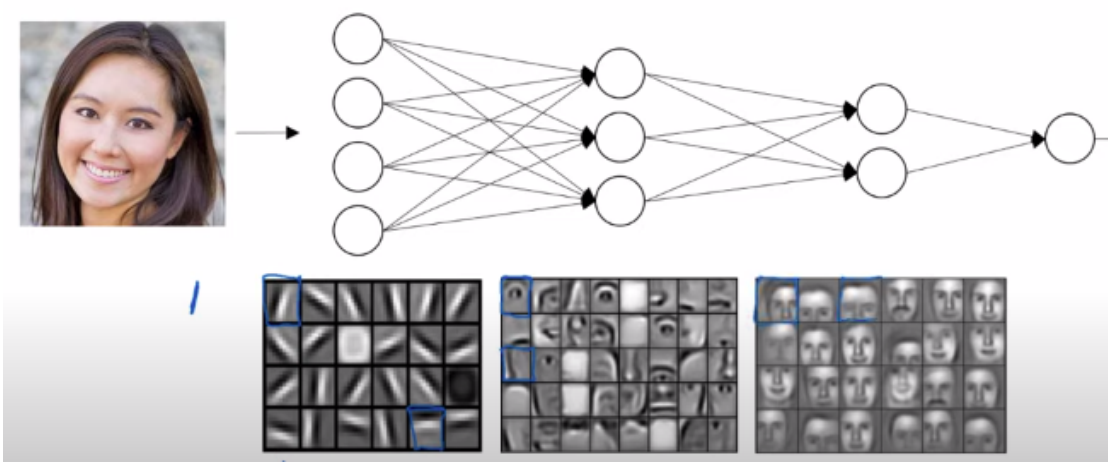
- **SVD**
- Sau khi thực hiện TF-IDF, chúng ta dễ dàng nhận thấy rằng, ma trận mà chúng ta thu được có kích thước rất lớn, và việc xử lý tính toán với ma trận này đòi hỏi thời gian và bộ nhớ khá tốn kém. Giả sử, chúng ta có *100.000* văn bản và bộ từ điển bao gồm *50000* từ, khi đó ma trận mà chúng ta thu được sẽ có kích thước là *100000 * 50000*. Giả sử mỗi phần

từ được lưu dưới dạng *float32* - 4 byte, bộ nhớ mà chúng ta cần sử dụng là:

  - $100000 \times 50000 \times 4 = 20000000000\ byte$

- Tức là chúng ta tốn tầm *18.63GB* bộ nhớ, khó có thể lưu hết vào RAM để thực hiện tính toán. Trong thực tế, với số lượng văn bản khổng lồ và từ điển lên đến hàng trăm nghìn từ, bộ nhớ mà chúng ta sử dụng còn tốn kém hơn rất nhiều. Để xử lý vấn đề này, chúng ta sẽ sử dụng thuật toán *SVD (singular value decomposition)* nhằm mục đích giảm chiều dữ liệu của ma trận mà chúng ta thu được, mà vẫn giữ nguyên được các thuộc tính của ma trận gốc ban đầu.

- **Deep Neural Network**
  - Tiếp theo chúng ta sẽ xây dựng một mạng Neural Network để phân loại cho các văn bản.
    Trong các mạng lưới thần kinh sâu, chúng ta có một số lượng lớn các lớp ẩn. Những lớp ẩn này thực sự đang làm gì? Để hiểu điều này, hãy xem xét hình ảnh dưới đây:

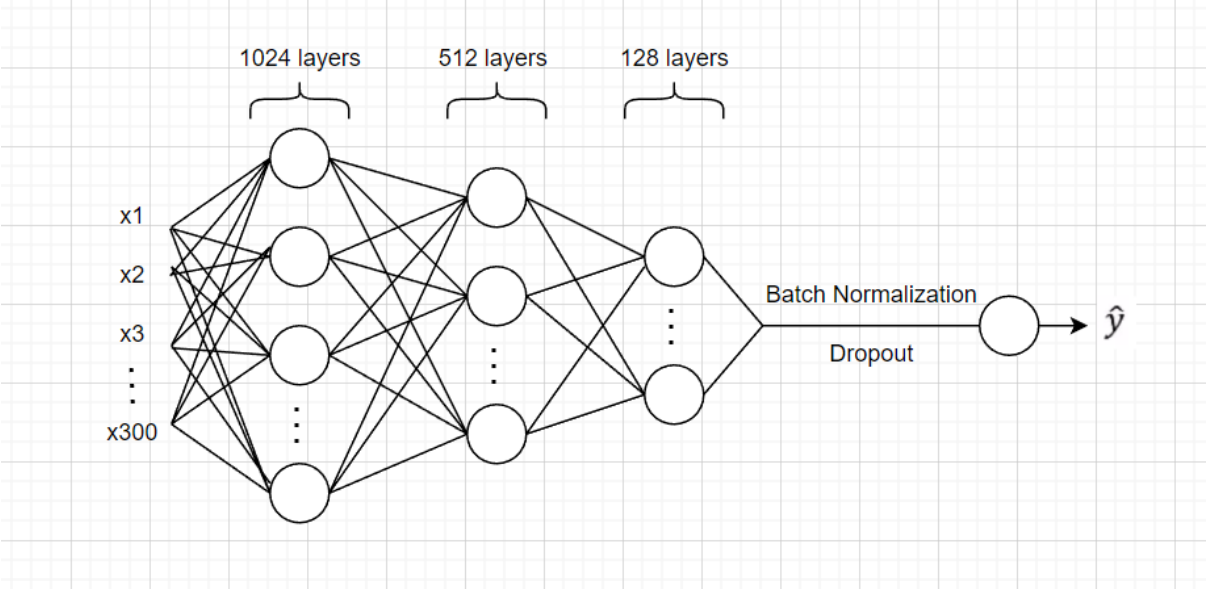


  - Mạng lưới thần kinh sâu tìm mối quan hệ với dữ liệu (quan hệ đơn giản đến phức tạp). Lớp ẩn đầu tiên có thể đang làm gì, đang cố gắng tìm các hàm đơn giản như xác định các cạnh trong ảnh trên. Và khi chúng ta đi sâu hơn vào mạng, các chức năng đơn giản này kết hợp với nhau để tạo thành các chức năng phức tạp hơn như nhận diện khuôn mặt. Một số ví dụ phổ biến về việc tận dụng Deep Neural Network là:
    - Nhận dạng khuôn mặt

        - Image ==> Edges ==> Face parts ==> Faces ==> desired face
    - Nhận dạng âm thanh:
        - Audio ==> Low level sound features like (sss, bb) ==> Phonemes ==> Words ==> Sentences
- Vậy tại sao lại là mạng DNN mà không phải cái gì khác?
  Ưu điểm:
  - Mạng NN có thể dùng cho cả dạng bài toán tuyến tính và bài toán phân lớp, có thể phù hợp với mọi loại dữ liệu số, bởi vì mô hình này là một mô hình toán học với phương trình xấp xỉ.
  - Mạng NN hoạt động tốt với dữ liệu phi tuyến tính với nhiều đầu ra. Nó đáng tin cậy trong cách tiếp cận các nhiệm vụ liên quan đến nhiều tính năng. Nó hoạt động bằng cách chia vấn đề phân loại thành một mạng lưới nhiều lớp gồm các phần tử đơn giản hơn.
- Tuy nhiên nó cũng có những nhược điểm:
    - NN là một hộp đen, chúng ta không thể biết được cách các biến tác động đến nhau.
    - Đây là một phương pháp rất tốn tài nguyên tính toán.
    - NN phụ thuộc rất nhiều vào dữ liệu, khi tập dữ liệu không đủ có thể dẫn đến hiện tượng overfitting và ảnh hưởng đến độ chính xác.

Tiếp theo chúng ta sẽ tự xây dựng mạng NN với kiến trúc như sau:

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 300)]             0

 dense_4 (Dense)             (None, 1024)              308224

 dense_5 (Dense)             (None, 1024)              1049600

 dense_6 (Dense)             (None, 512)               524800

 batch_normalization_1 (Batc  (None, 512)              2048
 hNormalization)

 dense_7 (Dense)             (None, 10)                5130

=================================================================
Total params: 1,889,802
Trainable params: 1,888,778
Non-trainable params: 1,024
_________________________________________________________________
```



- Mạng của chúng ta gồm có 3 lớp *Dense* (fully connected với số hidden layers lần lượt là 1024, 512, 128). Sau đó sẽ được đưa qua một lớp Batch Normalization và Dropout, cuối cùng cho ra được kết quả dự đoán.
- Trong mạng của chúng ta có sử dụng *Batch Normalization*. Batch Normalization là một phương pháp hiệu quả khi training một mô hình mạng nơ ron. Mục tiêu của phương pháp này chính là việc

muốn chuẩn hóa các feature (đầu ra của mỗi layer sau khi đi qua các activation) về trạng thái zero-mean với độ lệch chuẩn 1.

- Bên cạnh đó là dùng *Dropout*. Dropout là cách thức mà chúng ta giả định một phần các unit bị ẩn đi trong quá trình training, qua đó làm giảm tích hòa trộn (hay nói cách khác là 1 hidden unit không thể dựa vào 1 unit khác để sửa lỗi lầm của nó, dễ cho chúng ta thấy các hidden unit không đáng tin cậy).

7. Tuning Hyperparameters

- Nếu như Model parameter được mô hình sinh ra từ chính tập dữ liệu huấn luyện thì Model Hyperparameter lại hoàn toàn khác. Nó hoàn toàn nằm ngoài mô hình và không phụ thuộc và tập dữ liệu huấn luyện. Như vậy mục đích của nó là gì? Thực ra chúng có một vài nhiệm vụ như sau:
  - Được sử dụng trong quá trình huấn luyện, giúp mô hình tìm ra được các parameters hợp lý nhất
  - Nó thường được lựa chọn thủ công bởi những người tham gia trong việc huấn luyện mô hình
  - Nó có thể được định nghĩa dựa trên một vài chiến lược heuristics
- Chúng ta hoàn toàn không thể biết được đối với một bài toán cụ thể thì đâu là Model Hyperparameter tốt nhất và ta phải tự chọn qua kinh nghiệm. Chính vì thế trong thực tế chúng ta cần sử dụng một số kĩ thuật để ước lượng được một khoảng giá trị tốt nhất (Ví dụ như hệ số *k* trong mô hình k Nearest Neighbor. Sau đây mình xin đưa một vài ví dụ về Model Hyperparameter:
  - Chỉ số learning rate khi training một mạng nơ ron nhân tạo
  - Tham số *C* và sigma khi training một Support Vector Machine
  - Hệ số k trong mô hình k Nearest Neighbor
- Trong mạng NN tự thiết kế ở trên, mình đã chọn ra một số hyperparameters để điều chỉnh nhằm đem lại kết quả tốt nhất
  - Normalize: sử dụng hoặc không sử dụng lớp normalize, cụ thể trong bài là Batch Normalization.
  - Epoch: sử dụng early stopping để lấy được giá trị validation tại thời điểm hàm loss trên tập validation đạt kết quả tốt nhất (với độ kiên nhẫn là 5).

- Learning rate: scheduler (với step là Linear) trong quá trình fine tune/ training. Trong bài sẽ sử dụng hàm Exponential Decay với phương trình như sau:

  $learning\ rate = initial\ learning\ rate\ *\ decay\ rate^{(step/decay\ steps)}$

  Với lần lượt initial learning rate = 0.1

  decay rate = 0.96

  decay steps = 100000

- Optimizer: Stochastic Gradient Descent, Adam.

8. Các hàm đánh giá

Micro-average:

- MicroPrecision = $\frac{\sum_{c\in C} TP_c}{\sum_{c\in C} TP_c + FP_c}$
- Micro Recall = $\frac{\sum_{c\in C} TP_c}{\sum_{c\in C} TP_c + FN_c}$
- Micro-F-score = $\frac{2*MicroPrecision*Micro\ Recall}{MicroPrecision+MicroRecall}$

Trong đó TP_c, FP_c, FN_c lần lượt là TP, FP, FN của class c.

Macro-average:

- Macro Precision = $\frac{\sum_{c\in C} Precision_c}{|C|}$
- Macro Recall = $\frac{\sum_{c\in C} Recall_c}{|C|}$
- Macro-F-score = $\frac{2*Macro\ Precision*Macro\ Recall}{MacroPrecision+MacroRecall}$

9. Thực nghiệm và đánh giá

Mình đã chạy thực nghiệm trên bộ dữ liệu VNTC10 và VNTC27 trong 50 epochs, và lần lượt thêm/bớt các kỹ thuật tuning hyperparameters gồm Early Stopping, Scheduler Learning Rate, Add Batch Normalization, thay đổi Optimizer Function và cho ra kết quả như sau:

- **VNTC10**: 50 epochs - learning rate = 0.001

Model	**Epochs**	**Micro F1 score**	**Macro F1 score**
Base	50	0.84	0.80
Tuned	28	0.91	0.80
No scheduler - learning rate = 0.005	28	0.91	0.79
No Batch Normalization	7	0.14	0.02
SGD	37	0.91	0.79

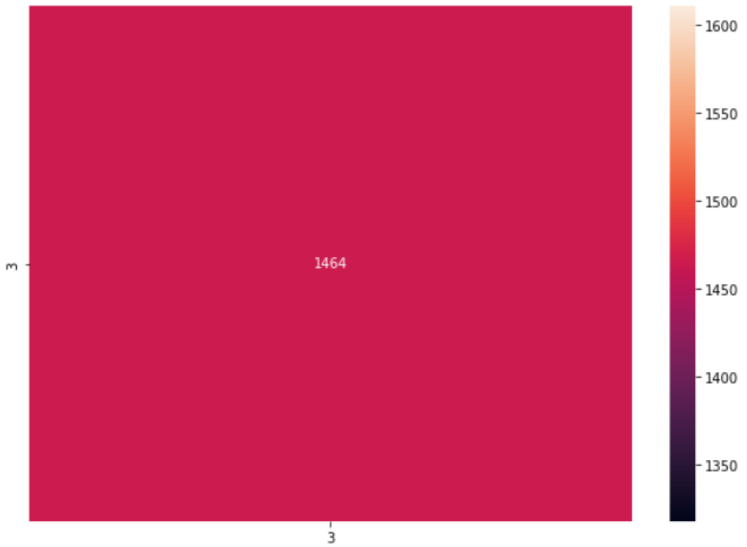
- **VNTC27**

Model	**Epochs**	**Micro F1 score**	**Macro F1 score**
Base	50	0.80	0.72
Tuned	28	0.83	0.74
No scheduler - learning rate = 0.005	43	0.88	0.79
No Batch Normalization	7	0.13	0.01
SGD	50	0.83	0.75

- Đối với bộ dữ liệu *VNTC10*, có vẻ mạng neural chúng ta thiết kế cho độ chính xác tối đa là 0.9 nên khi tuning các hyperparameters thì đội chính xác không xê dịch quá nhiều (<0.01). Tuy vậy cũng đã cải thiện được so với khi không tuning. Các phương pháp tuning cũng làm giảm đáng kể về thời gian training của mạng, cụ thể với phương pháp *Early Stopping* ta có thể giảm đến 22/50. Khi

dùng hàm *SGD* thay cho *Adam* thì số epochs cũng tăng lên một chút nhưng độ chính xác lại giảm đi một chút.

- Đối với bộ dữ liệu *VNTC27*, các chủ đề liên quan khá gần với nhau đã làm giảm độ chính xác của mạng đi gần 0.02. Bên cạnh đó, hiệu quả của các phương pháp tuning cũng tương tự với bộ VNTC10.
- Đặc biệt nhất, khi bỏ đi lớp Batch Normalization thì độ chính xác của mô hình giảm đi nghiêm trọng (còn khoảng 0.14) vì mô hình đã dự đoán toàn bộ tin tức thuộc cùng 1 class (hình bên dưới)



- **So sánh với các mô hình khác:**

  VNTC10:

Method	Micro F1 score	Macro F1 score
TF-IDF+DNN*	0.9078	0.7934
TF-IDF+BiLSTM	0.8696	0.8408
Doc2Vec+LSTM	0.8471	0.8169
GPT	0.9671	0.9664
GPT2	0.3612	0.7631
BART	0.8920	0.8517

VNTC27:

Method	Micro F1 score	Macro F1 score
TF-IDF+DNN*	0.8672	0.8011
TF-IDF+BiLSTM	0.8217	0.7897
Doc2Vec+LSTM	0.7667	0.7165
GPT	0.9431	0.8395
GPT2	0.6951	0.6343
BART	0.8724	0.8449

Mô hình TF-IDF + DNN có kết quả tốt hơn hầu hết các mô hình khác trong bài toán phân loại, nó chỉ kém hơn so với mô hình GPT bởi vì GPT đã được pretrained trên 1.3 tỷ dữ liệu tin tức. Vì các dữ liệu này cùng loại với dữ liệu dùng để test nên cho ra kết quả outperform so với các mô hình còn lại.

10. References

[1] Le Nguyen Nguyen, Cong Duy Vu Hoang, & Quoc Hung Ngo. (2019, January 17). ... ... - YouTube. Retrieved November 26, 2022, from https://ieeexplore.ieee.org/document/4223084/

[2] Nguyen Thanh Hau. (n.d.). *Phân loại văn bản tự động bằng Machine Learning như thế nào?* Viblo. Retrieved November 26, 2022, from https://viblo.asia/p/phan-loai-van-ban-tu-dong-bang-machine-learning-nhu-the-nao-4P856Pa1ZY3

[3] Simha, A. (2021, October 6). *Understanding TF-IDF for Machine Learning*. Capital One. Retrieved November 26, 2022, from

https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/

[4] Vu Huu Tiep. (2017, June 7). *Machine Learning cơ bản*. Machine Learning cơ bản. Retrieved November 26, 2022, from https://machinelearningcoban.com/2017/06/07/svd/