Nhận diện cảm xúc giọng nói (Speech Emotion Recognition Machine Learning)

I. TRỪU TƯỢNG

Một mô hình áp dụng học máy phân tích phân loại cảm xúc qua giọng nói được triển khai. Mục tiêu là phân tích âm thanh lời nói và phân loại cảm xúc tương ứng. Được sử dụng cho bất kỳ dự án nhận dạng dựa trên âm thanh như giọng nói, âm nhạc, bài hát,…

Đối với Train of the Model, những cảm xúc được thể hiện của các diễn viên đã được sử dụng từ cơ sở dữ liệu bộ bài phát biểu cảm xúc ở Toronto (TESS) từ Đại học Toronto được đào tạo và so sánh bởi thuật toán deep learning Long short-term memory (LSTM) và thuật toán máy học cơ bản K-Nearest Neighbors (KNN).

Song song đó xét đến tính bảo mật dữ liệu dạng âm thanh vì có thể chứa các nội dung nhạy cảm, nội dung Học Cộng Tác đã được triển khai theo phương pháp deep learning cụ thể là LSTM.

II. GIỚI THIỆU

Nghiên cứu này nhằm mục đích thực hiện cũng như tích hợp vào mô hình Trí tuệ nhân tạo (AI) sẽ phân tích đầu vào âm thanh lời nói trong thời gian thực, xác định và trình bày cảm xúc được thể hiện trong đó. Điều này có thể kết hợp làm tăng dữ liệu đầu vào cũng như tăng cơ sở để phân tích từ đó đưa qua quyết định và hướng xử lý qua lời nói trong thời gian thực hoặc qua file âm thanh

III. CÔNG CỤ VÀ PHƯƠNG PHÁP

1. Data

Bộ dữ liệu gồm 200 từ mục tiêu được nói trong cụm từ vận chuyển "Hãy nói từ _' bởi hai nữ diễn viên (26 tuổi và 64 tuổi) và các đoạn ghi âm được thực hiện từ bộ này mô tả từng cảm xúc trong số bảy cảm xúc (giận dữ, ghê tởm, sợ hãi, hạnh phúc , ngạc nhiên thú vị, buồn bã và trung lập). Tổng cộng có 2800 điểm dữ liệu (tệp âm thanh).

Tập dữ liệu được tổ chức sao cho mỗi diễn viên nữ và cảm xúc của họ được chứa trong thư mục riêng. Và trong đó, có thể tìm thấy tất cả 200 tệp âm thanh từ mục tiêu. Định dạng của tệp âm thanh là định dạng WAV.

2. Thuật toán sử dụng

- Long short-term memory (LSTM)

- Mô hình phân loại này được phát triển theo phương pháp deep learning, có nghĩa là một mạng lưới thần kinh sâu (DNN) trong khi một mô hình nâng cao để phân tích chuỗi các mẫu đã được chọn, đây là bộ nhớ ngắn hạn dài (LSTM).

- K-Nearest Neighbors (KNN)

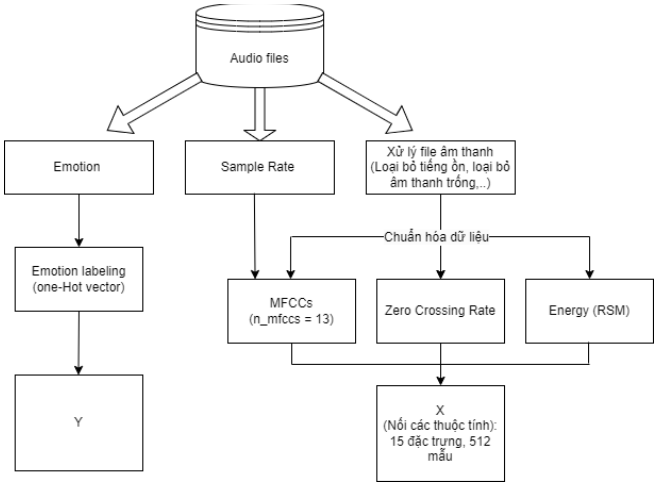
- Một thuật toán học máy có giám sát, đơn giản và dễ triển khai. Thường được dùng trong các bài toán phân loại và hồi quy được sử dụng. Thuật toán không dựa trên tập dữ liệu mà dựa trên sự phân bố của nhãn, nhãn của một đối tượng dữ liệu được dự đoán dựa trên phân bố của k nhãn xung quanh gần nó nhất.

IV. CHI TIẾT THỰC HIỆN

A. TIỀN XỬ LÝ DỮ LIỆU

a. Dữ liệu sau được trích xuất từ mỗi tệp âm thanh:

TESS: Tên tệp chứa chuỗi biểu thị cảm xúc, ví dụ: 'hạnh phúc'. Tỷ lệ mẫu: số lượng mẫu âm thanh mỗi giây cơ sở dữ liệu TESS được ghi ở 24,414kHz:



b. File âm thanh được xử lý theo thứ tự sau:

- Phiên bản 'AudioSegment': Âm thanh được tải vào một đối tượng bằng mô-đun AudioSegment của pydub.
- Chuẩn hóa: Đối tượng 'AudioSegment' được chuẩn hóa thành + 5,0 dBFS, bằng mô-đun của pydub.
- Chuyển đổi đối tượng thành một mảng mẫu bằng numpy & AudioSegment.
- Cắt khoảng trống ở đầu và cuối bằng librosa.
- Đệm mỗi tệp âm thanh đến độ dài tối đa bằng numpy, để cân bằng độ dài.
- Giảm tiếng ồn đang được thực hiện bằng noisereduce.

c. Chuẩn hóa dữ liệu và trích xuất tính năng:

Các tính năng được chọn đang được trích xuất bằng librosa cho mô hình nhận dạng cảm xúc giọng nói là:

- Energy - Root Mean Square (RMS)
- Zero Cross Rate (ZCR)
- Mel Frequency Cepstral Coefficient (MFCC)

(Với frame_length = 2048, hop_lentgh = 512, đảm bảo độ dài dãy bằng nhau.)

Kết hợp tất cả các tính năng với một biến 'X' duy nhất.

d. Mã hóa nhãn:

- Nhãn được trích từ tên tệp và chuyển dữ liệu từ dạng chữ sang số (Neutral:01, Happy:03,….)
- Điều chỉnh 'Y' với hình dạng 2D
- Chuyển đổi y_train và y_validation thành vectơ 'One-hot' cho mục đích phân loại

e. Sử dụng dữ liệu cho quá trình đào tạo mô hình

Tách X, Y thành các tập huấn luyện và kiểm tra với tỉ lệ:

- 80% bộ dữ liệu thành tập huấn luyện
- 20% bộ dữ liệu để đánh giá mô hình

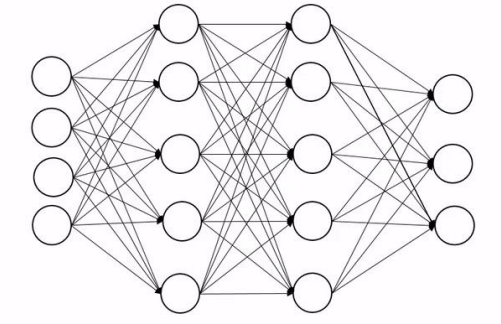
B. CHI TIẾT TRIỂN KHAI VÀ THUẬT TOÁN SỬ DỤNG

1. Machine Learinng

- LTSM

Khi triển khai Machine Learning thì trích xuất dữ liệu theo một tính năng Mel Frequency Cepstral Coefficient (MFCC) với Sample rate=13.

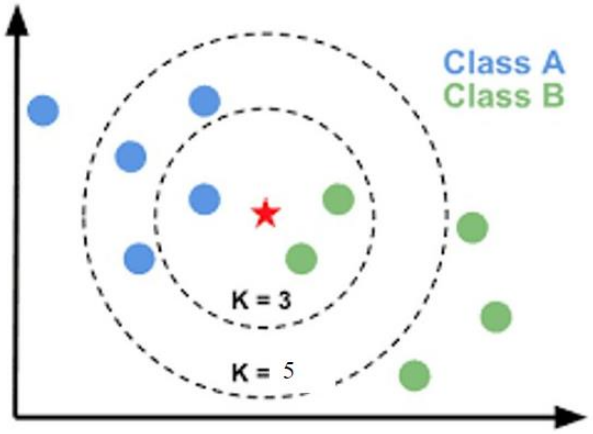
Mô hình được thực thi với thư viện Librosa, sử dụng 1 lớp LSTM ẩn với 256 nút và lớp đầu ra với 7 nút, mỗi lớp cho một cảm xúc bằng cách sử dụng kích hoạt 'softmax'. Trình tối ưu hóa dẫn đến kết quả tốt nhất là 'adam' với các tham số mặc định. Batchsize là 64, epoch: 50. Đầu vào mô hình là (2800,40,1) và Output là (2800,7).



Input layer : 40 audio frames
Hidden Layer Layer 1: LSTM 256 cells
Hidden Layer Layer 2: Dropout tỉ lệ 0.2, Kích hoạt Relu
Hidden Layer Layer 3: Dense 128 cells
Hidden Layer Layer 4: Dropout tỉ lệ 0.2, Kích hoạt Relu
Hidden Layer Layer 5: Dense 64 cells
Output layer : 7 node

- KNN

Triển khai mô hình KNeighbors với tham số k=5 và dạng đầu vào là (2800,40) và đầu ra (2800,7)



2. Federated Learning

Sử dụng Framework Flower cho Học cộng tác và dùng chiến lược giao tiếp FedAVG, khởi tạo num_clients là 2.

a) Client

Bộ dữ liệu

Khi triển khai Federated Learning trích xuất theo 3 tính năng đã nêu trên tiền xử lý dữ liệu và lưu thành hai file X_data.JSON và y_data.JSON.

Đào tạo mô hình

- Mô hình được thực thi với thư viện Librosa, sử dụng 2 lớp LSTM ẩn với 256 nút và lớp đầu ra với 7 nút, mỗi lớp cho một cảm xúc bằng cách sử dụng kích hoạt 'softmax'. Trình tối ưu hóa dẫn đến kết quả tốt nhất là 'RMSProp' với các tham số mặc định. Batchsize là 23, epoch: 200. Đầu vào mô hình là (2800,257,15) và Output.shape là (2800,7).

Input layer : 257 audio frames
Hidden Layer Layer 1: LSTM 64 cells
Hidden Layer Layer 2: LSTM 64 cells
Output Layer : 7 node

Đánh giá mô hình

- Sau khi mô hình đã được đào tạo để đáp ứng một số quan điểm thông qua quy trình FL, nó đã sẵn sàng để được đánh giá. Lưu ý rằng: tập thử nghiệm của thiết bị mới có sự phân bổ khác với tập huấn luyện nói chung.

b) Server

Khởi tạo mô hình

- Máy chủ chịu trách nhiệm khởi tạo các trọng số của mô hình ban đầu. Sau khi hoàn thành, mô hình ban đầu được chia sẻ với tất cả các máy khách và quá trình đào tạo có thể bắt đầu. Điều đáng chú ý là mỗi khách hàng bắt đầu với cùng một mô hình.

Tổng hợp mô hình

- Chúng ta cần cho framework biết cách xử lý/tổng hợp các chỉ số (metrics) tùy chỉnh này và chúng ta làm như vậy bằng cách chuyển các hàm tổng hợp chỉ số cho chiến lược(strategy). Sau đó, chiến lược sẽ gọi các chức năng này bất cứ khi nào nó nhận được số liệu phù hợp hoặc đánh giá từ khách hàng. Hai chức năng có thể là fit_metrics_aggregation_fn và evaluate_metrics_aggregation_fn .
- Sau khi nhận được các tham số mô hình cập nhật của mỗi máy khách,{ w_k :∀k ∈[K]}. Sử dụng trọng số trung bình để

tổng hợp mô hình.Với cách tiếp cận trung bình đường cơ sở, công thức cho điều này được đưa ra bởi công thức $w := \sum_{k=1}^{K} \frac{1}{K} w_k$.

C. KẾT QUẢ THỰC NGHIỆM

1. Machine Learning

Độ chính xác

KNN

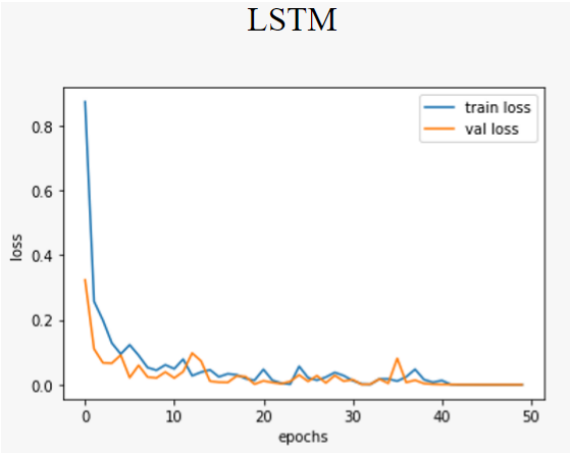
Accuracy	Val_acc
0,99	0,98

- Đối với LSTM, sau 50 vòng epoch thì độ chính xác đạt được là tiệm cận với. LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là khả năng của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.
- Đối với KNN, độ chính xác đạt được xấp xỉ 0.99.

=> Kết quả cho thấy cả hai mô hình đều cho ra độ chính xác khá cao dù đã được xử lý overfitting, cho thấy độ khả quan và phù hợp của mô hình với bộ dữ liệu.

=> Tương tự với độ mất mát của cả hai mô hình là rất nhỏ và khi so sánh độ chênh lệch giữa dự đoán và thực tế là rất thấp của cả hai mô hình.

Độ mất mát



2. Federated Learning

Round	Client 1				Client 2			
	Accuracy	Loss	Val_acc	Val_loss	Accuracy	Loss	Val_acc	Val_loss
1	0,7832	0,538	1,5496	0,5284	0,6224	1,0058	0,4898	1,2507
2	0,8776	0,3136	0,5306	1,5193	0,5510	1,4470	0,6020	1,1503
3	0,917	0,1199	0,6429	1,6387	0,8138	0,5626	0,6224	1,1947
4	1,0	0,01	0,6633	1,632	0,8010	0,5646	0,5918	1,1959
5	0,9923	0,0388	0,7041	1,512	0,6122	1,1172	0,4898	1,307

Qua từng vòng lặp ta nhận thấy được sai số và độ chính xác được cải thiện rõ rệt, tỉ lệ thuận với số vòng lặp

Cụ thể:

- Đối với LSTM: Với Client 1 sai số là 0.538, độ chính xác 0.7832 tương tự Client 2: 1.5008 và 0.6224 đối với mô hình huấn luyện ở vòng 1. Nhưng sau 5 vòng huấn luyện Client 1 có sai số chỉ còn 0.0388 độ chính xác tận 0.9923 tương tự Chient 2: 1.1 và 0.6
- Tuy rằng ở vòng cuối cùng kết quả của Client 2 không cho ra được chỉ số khả quan, nhưng qua cả năm vòng thì bộ tham số tối ưu cho Client 2 là ở vòng thứ 3 và Client 1 là vòng thứ 4. Và có thể xem xét đến việc tăng số vòng để có thể tìm ra một bộ tham số tối ưu hơn cho Client 2 đồng thời sử dụng thêm lớp ẩn Dropout để tránh bị overfitting.

=> Từ đây ta có thể kết luận: Mô hình học máy liên kết tổng hợp mô hình bằng phương thức trung bình liên kết (FedAvg) lấy trung bình các trọng số cho kết quả rất tốt.

Tính trung bình liên kết (FedAvg) cho phép các nút cục bộ thực hiện nhiều lần cập nhật hàng loạt trên dữ liệu cục bộ và trao đổi các trọng số được cập nhật thay vì độ dốc. Ngoài ra, việc lấy trung bình các trọng số được điều chỉnh đến từ cùng một lần khởi tạo không nhất thiết làm ảnh hưởng đến hiệu suất của mô hình được lấy trung bình. Tận dụng tối đa ưu điểm chính của việc sử dụng các phương pháp liên kết để học máy đảm bảo quyền riêng tư hoặc bí mật dữ liệu. Thật vậy, không có dữ liệu cục bộ nào được tải lên bên ngoài, nối hoặc trao đổi. Vì toàn bộ cơ sở dữ liệu được phân đoạn thành các bit cục bộ, điều này khiến việc xâm nhập vào cơ sở dữ liệu trở nên khó khăn hơn.

REFERENCES

https://github.com/pytorch/examples/tree/main/mnist
https://github.com/adap/flower/tree/main/examples/advanced_tensorflow
https://colab.research.google.com/github/tensorflow/federated/blob/v0.40.0/docs/tutorials/building_your_own_federated_learning_algorithm.ipynb#scrollTo=Yfld4oFNLo8Y
https://github.com/MeidanGR/SpeechEmotionRecognition_Realtime/blob/main/2_model.ipynb