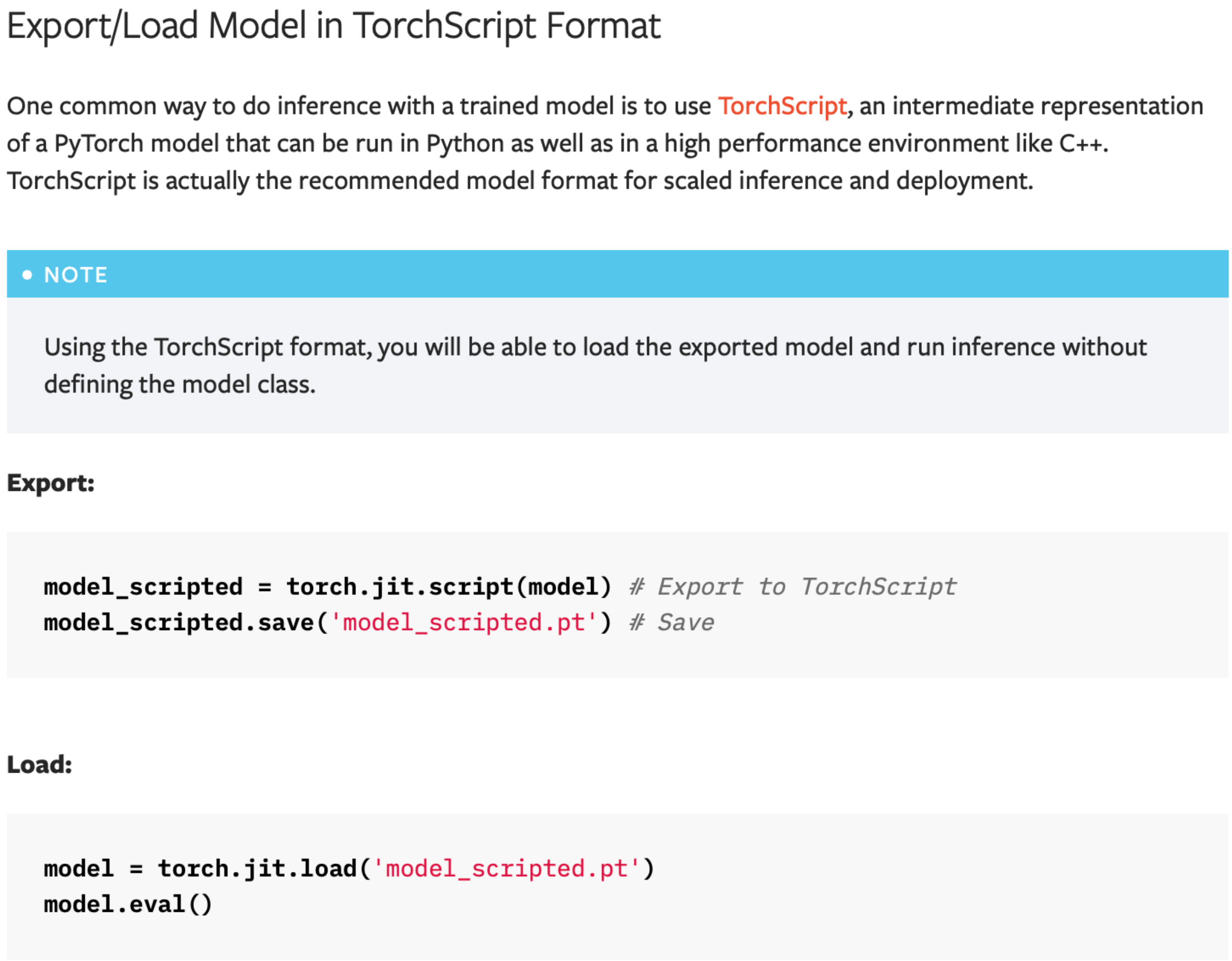
Karthik Raj
11/2/2023

Convolutional Neural Network (CNN) Data Preprocessing

Current Approach:

1. Inputs:
    a. Individual's Wav File with a sampling rate of 22,050
    b. Corresponding TextGrid file for individual which has same Wav filename except for extension (.Wav vs .TextGrid)
    c. Gender of Individual (Male or Female):
        i. Each gender has…
            1. Unique set of hyperparameters used to generate Mel Spectrograms (Feature used in classifying between intoxicated & sober)
            2. Intrinsic model & model's hyperparameters used to generate chunk level predictions
                a. Model is written in Torchscript format as recommended by PyTorch Devs for scaled deployment & inference
                b. Referred Documentation placed below…



Export/Load Model in TorchScript Format

One common way to do inference with a trained model is to use TorchScript, an intermediate representation of a PyTorch model that can be run in Python as well as in a high performance environment like C++. TorchScript is actually the recommended model format for scaled inference and deployment.

- NOTE

Using the TorchScript format, you will be able to load the exported model and run inference without defining the model class.

Export:

```
model_scripted = torch.jit.script(model) # Export to TorchScript
model_scripted.save('model_scripted.pt') # Save
```

Load:

```
model = torch.jit.load('model_scripted.pt')
model.eval()
```

Workflow:

2. Create Spectrify Object whose input parameters vary depending on user is male or female
    - a. Spectrify class accomplishes 2 key steps:
        - i. Extracts information of the **start and stop times of a chunk** that meets the **minimum time threshold of 1 second**
        - **ii.** Use that information to **create a predefined-size, mel spectrogram based on the gender-specific hyperparameters provided**
    - b. A chunk is essentially a sequence of phonetic sounds or utterances grouped together till the total time length of that chunk is at least 1 second long
        - i. TextGrids are used to identify when these phonetic sounds are occurring and primarily to remove silences and noise being present in the chunk
        - ii. This can be accomplished because the TextGrids provide an audio file mapping of when all phonetic sounds, long speech pauses, and noise occur
        - iii. To get a chunk of at least 1 second long, the start and stop times of phonetic sounds are sequenced together till the total time of the sequence is at least 1 second long
        - iv. The recorded output is the **start time of that chunk** along with **the end time of that chunk** and the **name of the file of the extracted chunk**
            1. Because these chunks are designed to meet a minimum time threshold of 1 second, additional processing needs to be done to create equally sized chunks which is the expected CNN input
3. Extract Chunk from Audio file whose start and end times are provided by the corresponding TextGrid of the Audio file
    - a. Relevant Spectrify object parameters are **silence length** (criteria used to determine if a person has stopped speaking or it's a small unavoidable pause between phonetic sounds which is okay) and **desired chunk length**
    - b. Relevant Spectrify Functions are planner and phraser
        - i. Planner extracts all phonetic sounds, noise, and pauses' start and stop times
        - ii. Phraser stitches them together with logic to output start and end time of the chunk that meet's minimum time threshold (1 second) with no noise or long pauses
    - c. The recorded output is the **start time of that chunk** along with **the end time of that chunk** and the **name of the file of the extracted chunk**
4. Create Mel Spectrogram:
    - a. TextGrid filename inputted alongside hyperparameters optimized for male or female participant
        - i. Implicit assumptions:
            1. .Wav file and .TextGrid file are same name except for extension
            2. Sampling Rate of .Wav file is 22,050
    - b. Mel Spectrogram is created with provided hyperparameters given during Spectrify object instantiation

  - c. Mel Spectrogram gets its intrinsic values normalized to be suitable for input to CNN
  - d. Mel Spectrogram gets its size truncated to a finite width
    - i. Height of Spectrogram is the number of mels which determines amount of detail in the frequency range
    - ii. Width of Spectrogram is cut to be strictly 1 second as determined by **int(Sampling Rate/ hop length) + 1**
  - e. Mel Spectrogram gets reshaped into an array of (num_channels, height of array, width of array) so (1, 64, 345) since 1 channel
  - f. All accomplished by a combination of the Spectrify's Spectrify function and a Pytorch Dataset object, which calls the Spectrify function and a function to reshape into a suitable format for the CNN

5. Create Predictions for each Chunk
   - a. Load in the male or female model (dependent on user's gender) in same fashion as PyTorch Documentation
   - b. Output of model will be logits essentially the result of the final dense layer's weights scaled by the output of previous dense layer + bias term
     - i. Output = Weights * Input(Output of Prev. Dense Layer) + Bias
   - c. Pass logits of each chunk into sigmoid function to generate pseudo probabilities: Values between 0 and 1
   - d. Predicted chunk class is 1 (Intoxicated) if probability >= 0.5. Otherwise, 0 (Sober)
6. Create User Wav File or User Sober/Intoxicated Prediction:
   - a. Each wav file can output 1 or more chunks so the resulting the wav file vote is the result of the aggregation of the chunk class predictions
   - b. If there is at least 1 chunk for each class:
     - i. Majority voting for user class prediction if number of chunks for a certain class is greater than the number of chunks for the other class
     - ii. If equal chunks for each class:
       - 1. Siding with positive class and outputting the mean of pseudo-probabilities by sigmoid function of chunks predicted positive
   - c. Else if only 1 unique chunk class (1 chunk or 1 set of chunks):
     - i. If sole class predicted is 1:
       - 1. User is predicted to be class 1 with mean of pseudo-probabilities by sigmoid function of chunks predicted positive
     - ii. If sole class predicted is 0:
       - 1. User is predicted 0
   - d. Output is User's Sober/Intoxicated Prediction:
     - i. 1 if Intoxicated and 0 if Sober