

## Round 2 Shortlisting Task Submission by Abhinav Tamta

### Text to Speech Converter with emotion , and Sentiment analysis model

The program performs two main tasks: text-to-speech synthesis with customised emotion like sad , happy and angry , audio generation and emotion prediction from text.

For the text-to-speech synthesis, the program utilizes pre-trained models (Tacotron2 and WaveGlow) from the NVIDIA/DeepLearningExamples repository. It loads these models and sets them up for inference on the GPU. The input text is provided, and the program generates mel-spectrograms using Tacotron2 and converts them into audio waveforms using WaveGlow. The resulting audio is saved as a WAV file and played back using the Audio class in the Jupyter Notebook.

For the speech to reflect emotions the pitch , the playback rate , audio levels have been manipulated.

For the emotion prediction from text, the program uses the VADER (Valence Aware Dictionary and sEntiment Reasoner) library. It installs the library and imports the SentimentIntensityAnalyzer class. The predict\_emotion function is defined, which takes the input text, calculates sentiment scores using the VADER analyzer, and predicts the emotion as "positive," "negative," or "neutral" based on the scores.

The program demonstrates the integration of these functionalities, showcasing how to synthesize speech from text and predict the associated emotion using pre-trained models and libraries.

Please note that the program includes additional code segments related to installing dependencies and loading utility functions, which may be necessary for the specific environment or use case but not directly related to the core functionality of text-to-speech synthesis and emotion prediction.

```
:  
import torch  
tacotron2 = torch.hub.load('NVIDIA/DeepLearningExamples:torchhub', 'nvidia_tacotron2', model_math='fp16')  
tacotron2 = tacotron2.to('cuda')  
tacotron2.eval()
```

```
waveglow = torch.hub.load('NVIDIA/DeepLearningExamples:torchhub', 'nvidia_waveglow', model_math='fp16')  
waveglow = waveglow.remove_weightnorm(waveglow)  
waveglow = waveglow.to('cuda')  
waveglow.eval()
```

## Bonus Functionality added

I have added a function to perform sentiment analysis , which tells wether the text prompt is positive , negative or neutral

- **In case of negative** : an emotion prompt of sad , angry can be used with it.
- **In case of positive**: happy prompt of emotion can be used

```
emotion = predict_emotion(text)
print("The emotion(sentiment) of the input text' ",text, "'is:", emotion)
```

The emotion(sentiment) of the input text' Radiant and joyful, I dance to the rhythm of happiness. 'is: positive

## Emotion Prompts

```
def modify_audio_emotion(input_file, output_file, emotion):
    # Load the input audio file
    audio = AudioSegment.from_file(input_file, format="mp3")

    # Modify the audio based on emotion
    if emotion == "happy":
        # Increase tempo by 10%
        audio = audio.speedup(playback_speed=1.2)
        # Increase pitch by 3 semitones
        audio = audio.set_frame_rate(int(audio.frame_rate * 1.1))

    elif emotion == "sad":
        # Apply a fade-out effect over the last 3 seconds
        fade_duration = 3000 # in milliseconds
        audio = audio.fade_out(fade_duration)
        audio = audio.set_frame_rate(int(audio.frame_rate * 0.8))
        audio= audio - 3

    elif emotion == "angry":
        # Increase tempo by 5%
        audio = audio.speedup(playback_speed=1.4)
        # Increase pitch by 2 semitones
        audio = audio.set_frame_rate(int(audio.frame_rate * 1.7))
        audio = audio + 7

    else:
        print("Invalid emotion specified.")
        return

    # Export the modified audio to the output file
    audio.export(output_file, format="mp3")
```

Here I have performed audio manipulation , there is space for improvement , but this was an original idea , which came to me while doing this task. In order to generate different emotions, I have modified the tempo , playback rate and applied fade effects.

The output videos are in the folder:

[https://drive.google.com/drive/folders/1fX-YHxXQvUyf\\_wVN9T4ImAsP1rbJ9K8U?usp=share\\_link](https://drive.google.com/drive/folders/1fX-YHxXQvUyf_wVN9T4ImAsP1rbJ9K8U?usp=share_link)

The output audio are in the folder:

<https://drive.google.com/drive/folders/1coCZROpds3Lj7XgHzg-wqU4yyKBBUVGz?usp=sharing>

the emotion specific files are

**tts\_angry ,tts\_sad ,tts\_happy**