Assignment 2 - ASR

Technical Part - Python3.10

Data Acquisition

1. Record 9 individuals (at least 4 females and 4 males) pronouncing the digits 0-9 plus a non-digit random word (e.g. “banana”). You can share your recordings with other students from the class.
    a. The recordings should be manually segmented so that each speaker has 11 audio files, each containing one of the digits from 0 to 9 and the random word. This results in a dataset of size 99.
    b. To ensure consistency, resample all audio files to a sampling frequency of 16 kHz.
    c. Once the recordings are segmented and resampled, divide the data into three groups:
        i. Class Representative: 1 individual chosen to represent the entire class.
        ii. Training Set: Consisting of 2 males and 2 females selected from the recorded speakers.
        iii. Evaluation Set: Consisting of the remaining speakers, with 2 males and 2 females.
2. For each audio file calculate the Mel Spectrogram (you can use librosa):
    a. Window size of 25ms
    b. Hop size of 10ms
    c. *N* filter banks: 80
    d. After computing the Mel Spectrogram, present several samples from the data and analyze the following:
        i. Differences within Speaker Samples (Different Digits)
        ii. Differences across Digit Samples (Different Speakers/Genders)

DTW

3. Implement dynamic time warping (DTW):
    a. Select the class representative recordings as the reference database (DB)
    b. Implement the DTW algorithm that was described in the lecture.
    c. **Compare each audio recording in the training set with each of the audios in the DB using DTW algorithm.** This algorithm aligns two sequences of feature vectors (in this case, Mel Spectrograms) by finding the optimal alignment that minimizes the total cost between them.
    d. Distance Matrix: Construct a distance matrix with dimensions 4x10x11. Where 4 is the number of speakers, 10 is the number of digits, and 11 is the number of

reference signals. Each cell in the matrix represents the DTW cost between a recording in the training set and a recording in the DB.

    i. You can present it as a 40x11 matrix, where 40 is the number of speakers times the number of digits.

e. Set Threshold and Determine Classification: Set up a threshold on the distances to determine the classification of each audio signal. Your algorithm should not label the random word (as it should be below the similarity threshold). Classify each recording in the training set based on the closest match in the DB. Calculate the accuracy over the training set.

f. Apply Threshold on Validation Set: Apply the selected threshold on the distance matrix of the validation set. Construct a confusion matrix to evaluate the classification accuracy over the validation set.

g. Try to improve results by:

    i. Normalizing the audio (using AGC)

    ii. Normalizing the distance w.r.t the length of each audio file

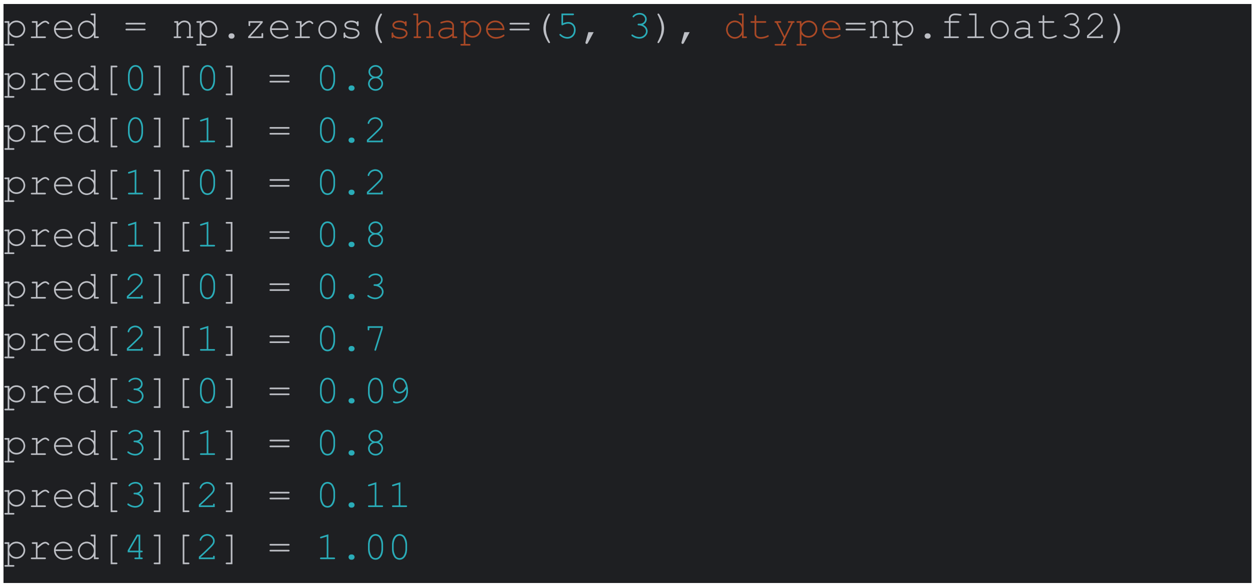
    iii. Plot the confusion matrix

Forward Algorithm

4. Implement the CTC’s collapse function B

5. Implement the forward pass (using the forward variable $\alpha_s(t)$) of the CTC algorithm presented in class (similar to the evaluation in HMM, without the transition probabilities).

    a. Given the following probability matrix:



```
pred = np.zeros(shape=(5, 3), dtype=np.float32)
pred[0][0] = 0.8
pred[0][1] = 0.2
pred[1][0] = 0.2
pred[1][1] = 0.8
pred[2][0] = 0.3
pred[2][1] = 0.7
pred[3][0] = 0.09
pred[3][1] = 0.8
pred[3][2] = 0.11
pred[4][2] = 1.00
```

    b. And the following alphabet label mapping: {0: 'a', 1: 'b', 2:’^’}, where ‘^’ is the blank symbol.

    c. What is the probability of the sequence ‘aba’?

    d. Plot the pred matrix:

        i. Add y labels according to the label mapping.

    ii. You can use log probs for calculation and plotting.

6. Adapt the forward pass of the CTC for force alignment (similar to the decoding in HMM, without the transition probabilities):
    a. Replace the ‘sum’ operator with the ‘max’ operator
    b. Find the most probable path for the sequence ‘aba’?
    c. What is the probability of that path?
    d. plot the aligned sequence (before collapse) on top of the prob_mat, and print the sequence labels.
    e. plot the backtrace matrix, and the selected path.
7. Repeat Q6 on the followings:
    a. Load the ‘force_align.pkl’ pickle file. This will load a dictionary with the following keys:

```
data = pkl.load(open('$PATH', 'rb'))
```

        i. Audio - the audio, sampled at 16KHz.
        ii. **Label mapping**- a dictionary between label (symbol) and index. The keys are the indices in acoustic_model_out_probs, and the values are the corresponding characters (alphabet + blank).
        iii. **acoustic_model_out_probs**- the acoustic model’s output (probability matrix). The dimensions of the matrix are $Tx29$, where T is the temporal length, and 29 is the alphabet size (including blank)
        iv. gt_text- the ground truth text.
        v. **text_to_align**- the label sequence (text) we want to force align.