

Examination Guidelines

1. Any kind of plagiarism is not accepted. We will strictly follow institute policies for plagiarism.
2. Programming languages: Python + PyTorch.
3. You may use any external libraries or GitHub codes. However, the evaluation will test your knowledge of the algorithm and the choice of hyperparameters. Do cite the libraries/codes.
4. **A single zip file containing the report, codes and readme if required. The zip file should be named Rollno_Minor2.zip.**

Question 1: Create an audio dataset of 100 examples for source separation such that source1 is a sinusoid with a random frequency smaller than fundamental frequency $f_s = 8000$, and source2 is a sinusoid with a frequency larger than f_s . The length of time steps (T) is 10000. Create a mixture signal from source1 and source2 such that mixture = source1 + source2. Divide the 100 examples in a 90:10 ratio for train and test.

- (i) Plot the source1 spectrogram, source2 spectrogram and mixture spectrogram. [2 Marks]
- (ii) Design a custom architecture using LSTM and linear layer for separating the source signal from the mixture signal. Train the model for ten epochs using l1 loss and report the SI-SNR value on the Test set. [5 Marks]
- (iii) Plot the spectrogram for reconstructed source1 and reconstructed source2. Compare this spectrogram with ground-truth spectrograms for source1 and ground-truth spectrograms for source2. [3 Marks]

Question 2: Connectionist Temporal Classification (CTC) loss is for training neural networks for ASR and to solve the missing alignments problem. Download the [HyperValley Dataset](#) to train a multi-task architecture for transcription and emotion classification. Split the dataset into 80:20 for train and test.

- (i) Implement the CTC loss from scratch. The loss function should take five following parameters : (a) log_probs: The log-beliefs returned by an ASR model, (b) targets: contiguous output labels (no blanks), (c) input_lengths: Lengths of each example in minibatch, (d) target_lengths: Lengths of each target in minibatch (e) blank: The token that is used to represent silence. [3 Marks]
- (ii) Using the data below, compare your loss function with [torch CTC loss](#) [1 Marks]

```
log_probs = torch.randn(50, 16, 20).log_softmax(2)
targets = torch.randint(1, 20, (16, 30), dtype=torch.long)
input_lengths = torch.full((16,), 50, dtype=torch.long)
target_lengths = torch.randint(10,30,(16,), dtype=torch.long)
blank = 0
```

- (iii) Design a custom multi-task architecture for an ASR task and emotion classification. Use CTC loss to train the ASR head and cross-entropy loss to train the emotion classification head. [6 Marks]