HW1-Bootcamp (HW1P2 Part)

January 17, 2026

TAs: En Zheng (enzheng@andrew.cmu.edu)
Mugur Preda (mpreda@andrew.cmu.edu)
Felix Hirwa Nshuti (fhirwans@andrew.cmu.edu)

Carnegie Mellon University

Homework submission timeline

- Checkpoint submission (Friday, 23th January 2026, 11:59 PM EST)

- Initial submission to Kaggle with a preliminary score of at least 60%
- Complete the checkpoint notebook and submit the .json file to Autolab

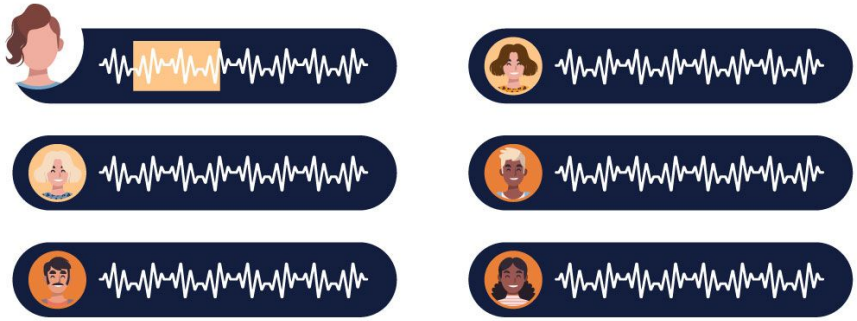
- On-time code submission (Friday, 6th February 2026, 11:59 PM EST)

- You need to improve your model performance to achieve the cutoff
- Then submit the code and related files to Autolab by 8th February 2026, 11:59 PM EST

- Slack submission (Friday, 13th February 2025, 11:59 PM EST)

- There is a slack Kaggle competition (join the competition via the link given in Piazza)
- Generate the submission.zip file and submit to the same Autolab as a late submission

Overview



- What to do?

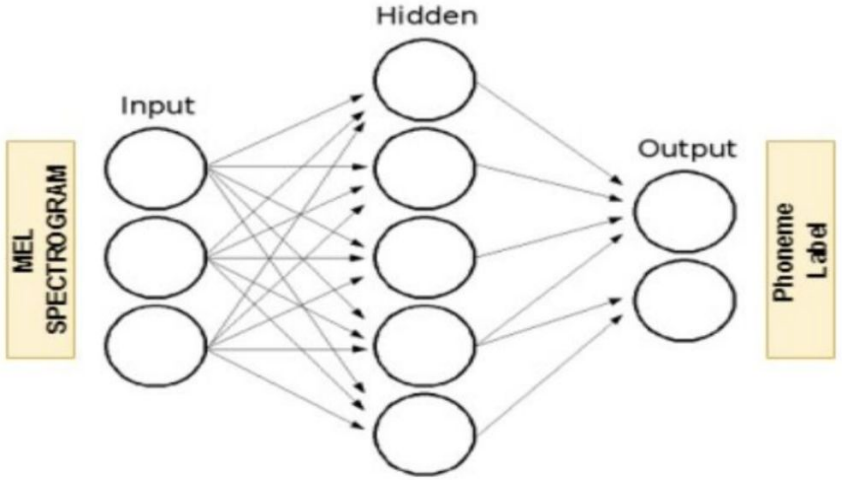
Speech Recognition: generate predictions for the phonemes of the test set.

- How to do?

Building your multilayer perceptron (MLP)!

Carnegie Mellon University

Overview



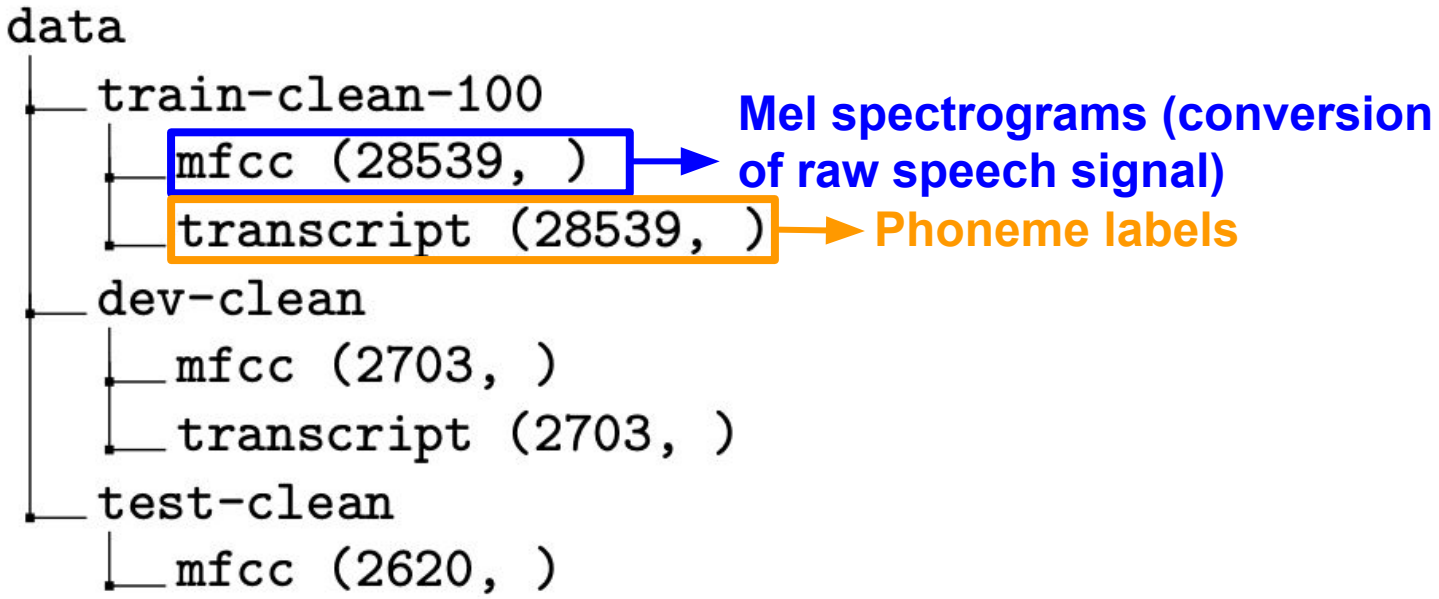
Dataset of Audio recordings (utterances)

MLP

Phoneme labels

Carnegie Mellon University

Dataset

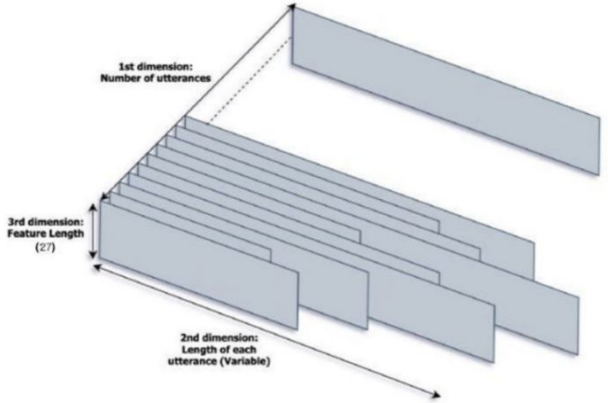


Audio representation

Raw Speech Signal
(Speech waveform)

Short Time
Fourier Transform

Melspectogram



Carnegie Mellon University

Transcripts

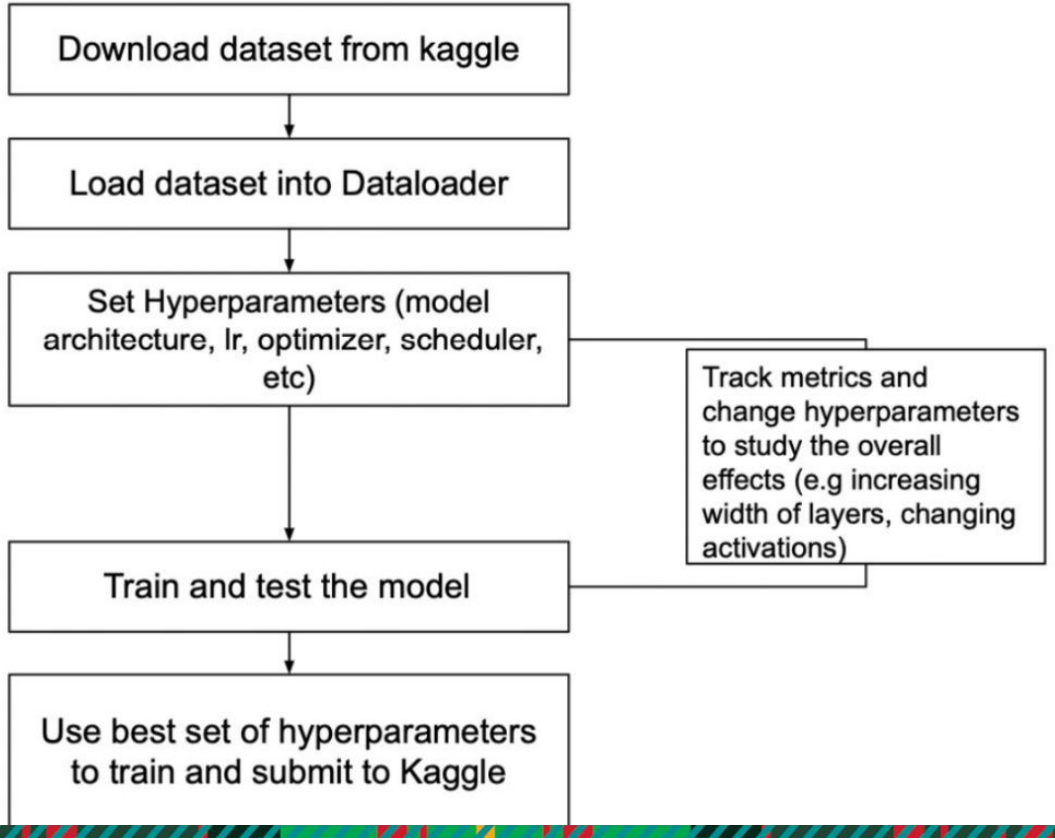
```
PHONEMES = [
            '[SIL]',   'AA',    'AE',    'AH',    'AO',    'AW',    'AY',
            'B',     'CH',    'D',     'DH',    'EH',    'ER',    'EY',
            'F',     'G',     'HH',    'IH',    'IY',    'JH',    'K',
            'L',     'M',     'N',     'NG',    'OW',    'OY',    'P',
            'R',     'S',     'SH',    'T',     'TH',    'UH',    'UW',
            'V',     'W',     'Y',     'Z',     'ZH',    '[SOS]', '[EOS]']
```

Dataset

```python
# Dataset class to load train and validation data

class AudioDataset(torch.utils.data.Dataset):

    def __init__(self, root, phonemes = PHONEMES, context=0, partition= "train-clean-100"): # Feel free to add more arguments

        self.context    = context
        self.phonemes   = phonemes
        self.subset = config['subset']

        # TODO: Initialize augmentations. Read the Pytorch torchaudio documentations on timemasking and frequencymasking
        self.freq_masking = NotImplemented
        self.time_masking = NotImplemented


        # TODO: MFCC directory - use partition to acces train/dev directories from kaggle data using root
        self.mfcc_dir       = NotImplemented
        # TODO: Transcripts directory - use partition to acces train/dev directories from kaggle data using root
        self.transcript_dir = NotImplemented

        # TODO: List files in sefl.mfcc_dir using os.listdir in SORTED order
        mfcc_names          = NotImplemented
        # TODO: List files in self.transcript_dir using os.listdir in SORTED order
        transcript_names    = NotImplemented

        # Compute size of data subset
        subset_size = int(self.subset * len(mfcc_names))

        # Select subset of data to use
        mfcc_names = mfcc_names[:subset_size]
        transcript_names = transcript_names[:subset_size]

        # Making sure that we have the same no. of mfcc and transcripts
        assert len(mfcc_names) == len(transcript_names)

        self.mfccs, self.transcripts = [], []


        # TODO: Iterate through mfccs and transcripts
        for i in tqdm(range(len(mfcc_names))):

            # TODO: Load a single mfcc. Hint: Use numpy
            mfcc            = NotImplemented
            # TODO: Do Cepstral Normalization of mfcc along the Time Dimension (Think about the correct axis)
            mfccs_normalized = NotImplemented

            # Convert mfcc to tensor
            mfccs_normalized = torch.tensor(mfccs_normalized, dtype=torch.float32)
```

Carnegie Mellon University

Workflow



Carnegie Mellon University

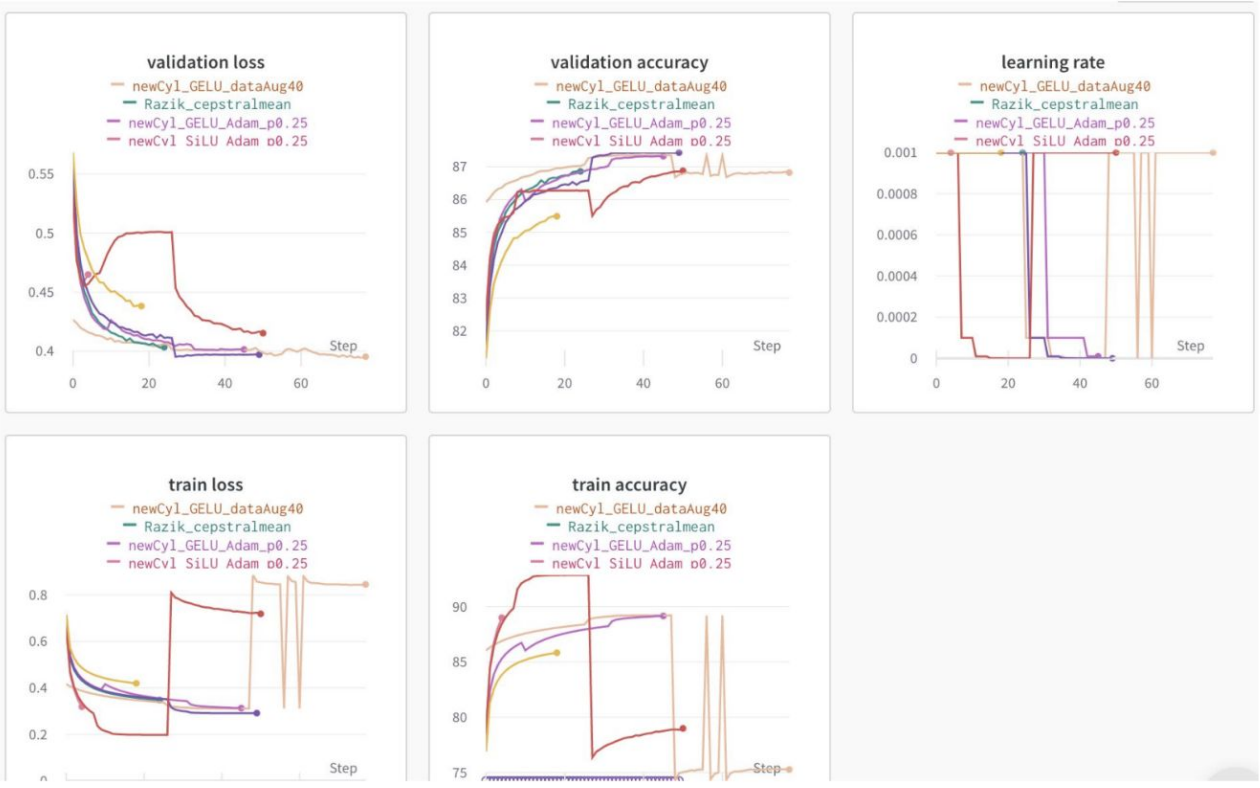
Running Ablations

Hyperparameters	Values
Number of Layers	2-8
Activations	ReLU, LeakyReLU, softplus, tanh, sigmoid
Batch Size	64, 128, 256, 512, 1024, 2048
Architecture	Cylinder, Pyramid, Inverse-Pyramid, Diamond
Dropout	0-0.5, Dropout in alternate layers
LR Scheduler	Fixed, StepLR, ReduceLROnPlateau, Exponential, CosineAnnealing
Weight Initialization	Gaussian, Xavier, Kaiming(Normal and Uniform), Random, Uniform
Context	0-50
Batch-Norm	Before or After Activation, Every layer or Alternate Layer or No Layer
Optimizer	Vanilla SGD, Nesterov's momentum, RMSProp, Adam
Regularization	Weight Decay
LR	0.001, you can experiment with this
Normalization	You can try Cepstral Normalization

Table 1: Hyperparameter Tuning

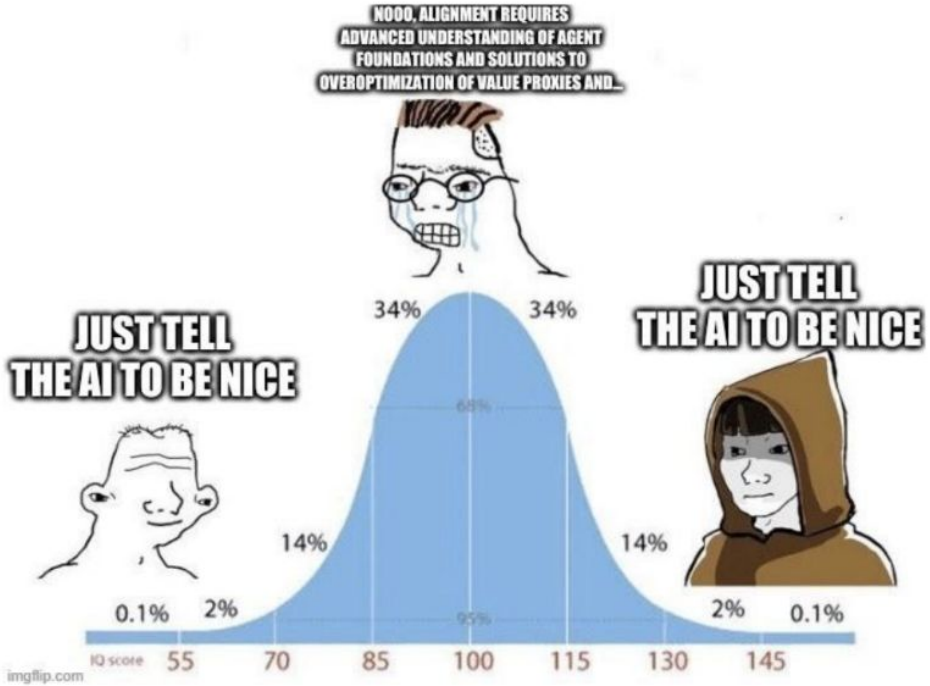
Carnegie Mellon University

Running Ablations



Carnegie Mellon University

Running Ablations



- Progressively build on your experiments
- Incorporate some domain knowledge
- Start with several simple architectures

Carnegie Mellon University