RNN Acoustic Models for Speech Recognition

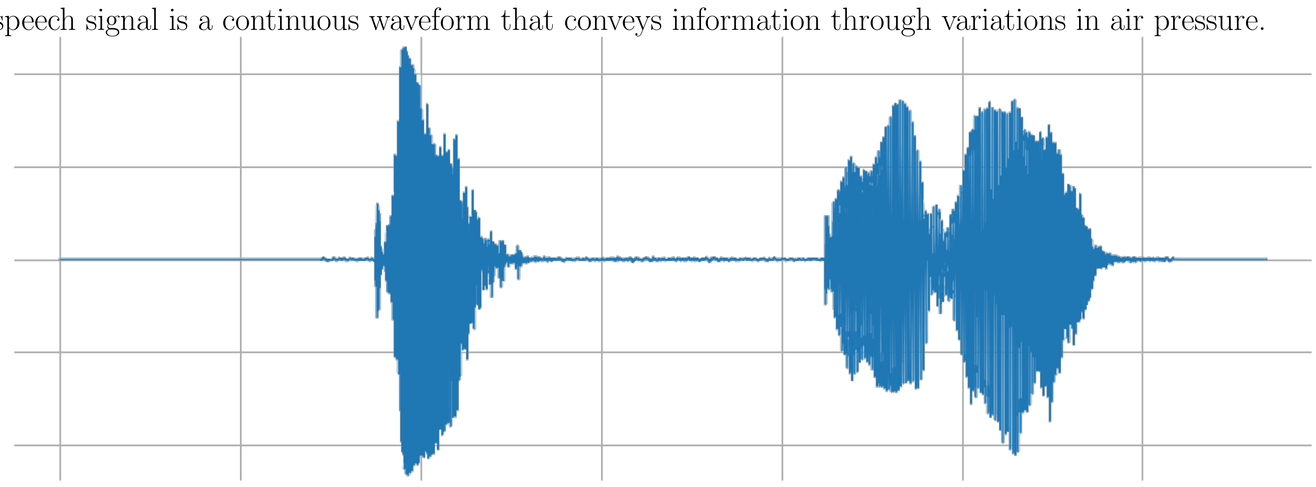
Ad882

Introduction to Speech Recognition

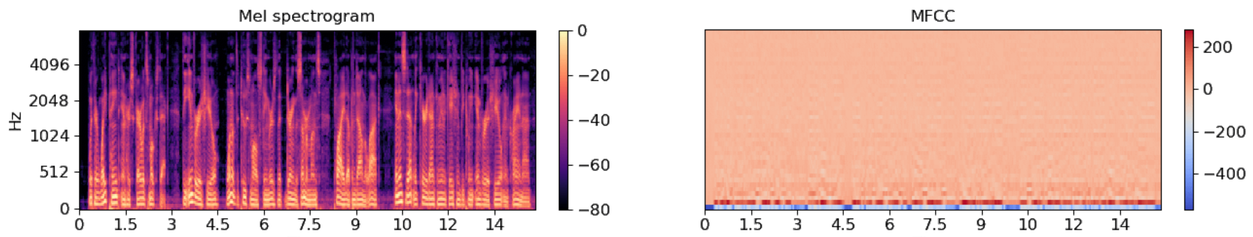
Speech Recognition, or Automatic Speech Recognition (ASR), converts spoken language into text using computational models. Unlike voice recognition (which identifies speakers) or emotion recognition (which detects sentiment), ASR focuses solely on transcription.
Traditional methods, such as Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs), have evolved towards Deep Neural Networks (DNNs). Given the sequential nature of speech, Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) models, are preferred for real-time speech recognition due to their efficiency in handling temporal dependencies.

I. Representing Speech Signals

A speech signal is a continuous waveform that conveys information through variations in air pressure.



Mel-Frequency Cepstral Coefficients (MFCCs) are widely used in ASR as they provide a compact and perceptually relevant representation of speech. Derived from the Mel spectrogram, they capture both spectral and cepstral features, making them highly effective for speech recognition.



MFCCs are preferred because they mimic human auditory perception, reduce dimensionality and enhance robustness.

II. Feature Processing

1. Data Augmentation

To improve robustness and generalization, audio data augmentation applies transformations to artificially expand the dataset. Common techniques include *adding noise*, *stretching time*, *shifting time*, *scaling volume*...

2. Extracting MFCCs

MFCCs are computed from speech signal $x(t)$ using the following steps:

- Framing: The signal is divided into 25ms frames with a 10ms stride.

$$x_w[m,n] = x[n+mH] \cdot w[n], \quad 0 \le n < N, \quad m = 0,1,2,\dots$$

- FFT & Power Spectrum: A Fast Fourier Transform extracts frequency components.

$$X(f) = \sum_{n=0}^{N-1} x_w(n) e^{-j2\pi fn/N}$$

- Mel Filterbank: Frequencies are mapped onto the Mel scale to mimic human.

$$S_{\text{mel}}(m) = \sum_{f=0}^{F-1} H_m(f) P(f)$$

- Logarithm & DCT: A log transformation and Discrete Cosine Transform (DCT) produce the final MFCCs, capturing essential speech features.

$$E_m = \log S_{\text{mel}}(m) = \log \sum_{f=0}^{F-1} H_m(f) P(f)$$

$$C_n = \sum_{m=0}^{M-1} E_m \cos\left(\frac{\pi n(2m+1)}{2M}\right)$$

3. Frame Stacking and Decimation

To optimize computation, multiple frames are stacked before skipping forward. This technique reduces computational cost while preserving speech information, provides a richer temporal context for ASR models and enhances generalization by capturing phonetic structures over time.

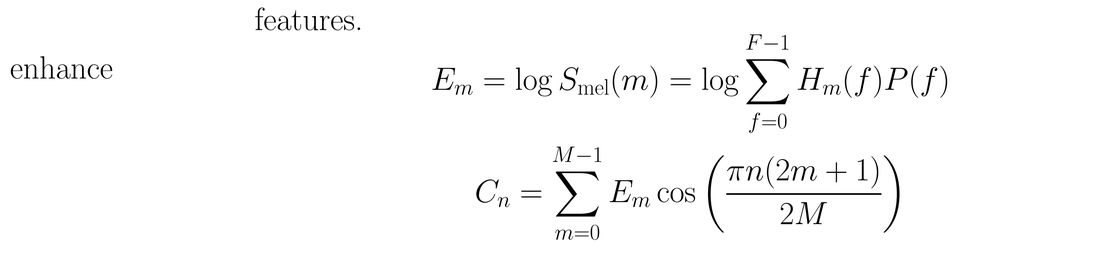
III. Encoding Audio Transcripts

The transcript must be encoded for use in the ASR model. Instead of passing full sentences, the most effective approach is to represent transcripts **phonematically**, using phonemes—the smallest units of sound in a language:

transcript $\rightarrow$ /t/ /r/ /ae/ /n/ /s/ /k/ /r/ /i : / /p/ /t/ $\rightarrow$ 22 5 24 27 9 40 5 38 17 22

IV. Model

The model is a **5-layer biderectional LSTM-based Recurrent Neural Network (RNN)**. Each LSTM layer consists of **300 neurons** per direction, resulting in a **600-dimensional hidden state** after concatenation of the forward and backward layers.



The output of the model is a sequence of **probability distributions** over the possible phonemes in the dictionary at each timestep. Formally, let T be the number of timesteps, and N the size of the phoneme dictionary. At each timestep t, the model predicts a vector of size N containing the probabilities for each possible phoneme:

$$P(l|x_t) = \begin{bmatrix} P(l_1|x_t) \\ P(l_2|x_t) \\ \vdots \\ P(l_N|x_t) \end{bmatrix}$$

In practice, the model computes **log-softmax**.

V. Connectionist Temporal Classification Loss

The challenge lies in the alignment problem: the model must map an input sequence of variable length to a target transcript. Unlike classic loss functions, which assume fixed alignments between inputs and outputs, CTC can handle this misalignment by introducing a special "blank" symbol ($\varnothing$).
Thus, multiple alignments π can correspond to the same target transcript y. For example, the word "hello" can be represented in different ways depending on the sampling: *h∅e∅l∅lo / hhh∅eeellllllooo ...*
Since explicit alignments are not provided, all valid alignments π that can generate the target y are summed:

$$P(y|x) = \sum_{\pi \in \mathcal{A}(y,T)} P(\pi|x)$$

The CTC loss function for the correct transcript y given the input x is :

$$\mathcal{L}_{CTC} = -\log P(y|x)$$