

B.Tech. Project Report

on

Automatic Speech Recognition using Deep Learning

Submitted by

Aashish Yadavally
(201451011)

under the supervision of

Dr. Anil Kumar Vuppula

Mentor:

Dr. Ajay Nath Jha

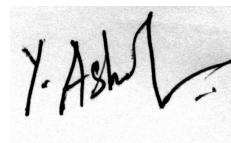


INDIAN INSTITUTE OF INFORMATION TECHNOLOGY VADODARA

2018

Declaration

I, Aashish Yadavally, declare that this written submission represents my ideas in my own words and where other's ideas or words have been included, I have adequately cited and referred the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated any idea/data/fact/source in my submission. I fully understand that any violation of the above will cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained.

A handwritten signature in black ink, appearing to read 'Y. Ashish', is written on a light gray background.

AASHISH YADAVALLY

Date: 27th April, 2018

Roll No. 201451011

Acknowledgement

I express my deepest thanks to our Director, Dr. Sarat Kumar Patra (Indian Institute of Information Technology - Vadodara) for allowing me to carry out the research internship and supporting me throughout. I would also like to express my sincere gratitude to my supervisor Dr. Anil Kumar Vuppula for providing me the opportunity to be a part of the Speech and Vision Lab at IIIT Hyderabad, and for his invaluable guidance and suggestions throughout the course of the project. I would like to specially thank my mentor, Dr. Ajay Nath Jha, for his feedback regarding the progress of my project. Also, I would like to specially thank Mr. Krishna Gurugubelli in the Speech and Vision Lab at IIIT Hyderabad for constantly motivating me to work harder, and for guiding me with the intricacies of the project and the Kaldi toolkit.

Abstract

Siri, Cortana, Google Now, are examples of speech recognition systems that are prominent today, which are changing the way digital personal assistants work. Well, speech recognition has been around for close to six decades now, but it has taken massive strides in terms of accuracy, since 2012, when four research groups teamed up, exploring neural networks for the acoustic modeling in speech recognition systems (overview of the research is presented in [1]). Now, deep learning models are further being explored and used in the speech recognition architectures, resulting in extensive research on Deep Learning in the context of Speech Recognition. End-to-end speech recognition systems are the state-of-the-art systems on which current ASR research is taking place, motivating me to pursue and study this technology in greater detail.

Contents

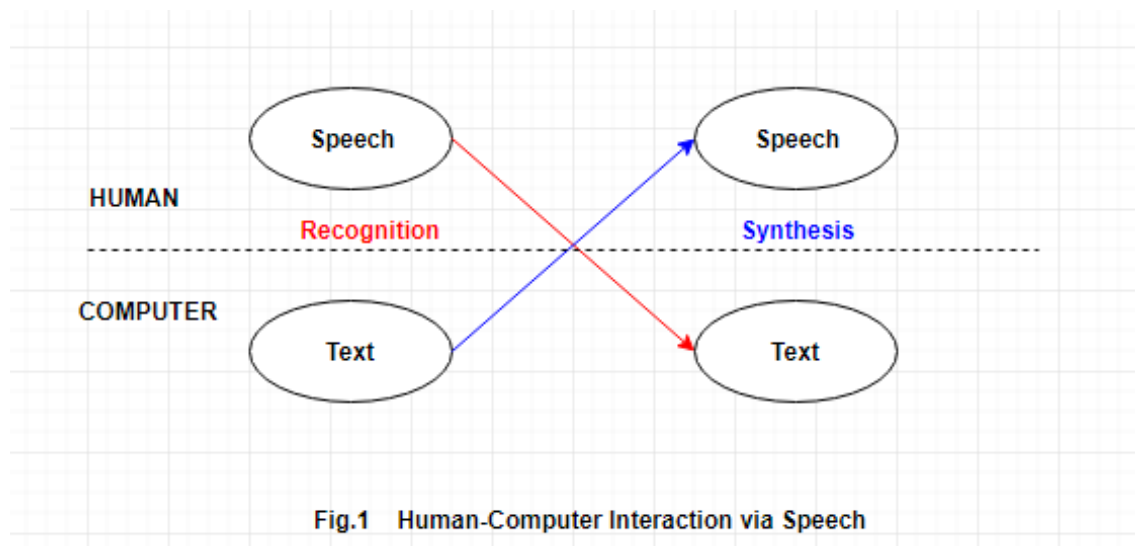
1	Introduction	1
1.1	Project Overview	2
1.2	Goals of the Project	3
2	Automatic Speech Recognition	4
2.1	Signal Pre-Processing and Feature Extraction	5
2.2	Acoustic Modeling	7
2.2.1	GMM-HMM Acoustic Modeling framework	7
2.2.2	DNN-HMM Acoustic Modeling framework	8
2.2.3	TDNN-HMM Acoustic Modeling framework	9
2.2.4	SGMM-HMM Acoustic Modeling framework	10
2.2.5	TDNN-LSTM Acoustic Modeling framework	10
2.2.6	RNN-LSTM Modeling framework	11
2.3	Language Modeling	12
2.4	Linguistic Decoding	13
3	Automatic Speech Recognition using Kaldi	14
3.1	Structure of the Kaldi Toolkit	14
3.2	Signal Processing and Feature Extraction in Kaldi	16
3.3	Acoustic Modeling in Kaldi	17
3.4	Language Modeling in Kaldi	17
3.5	Decoding in Kaldi	18

4	Experimentation	20
4.1	Data	20
4.2	Training	20
4.3	Results and Discussion	21
5	Conclusions and Future Work	25

Chapter 1

Introduction

The primary mode of communication among human beings is via speech. The constant aim of researchers has been to promote human-machine interaction and, speech recognition and speech understanding form the main components of this interaction. Automatic Speech Recognition is the process of deriving information from an audio/speech sample by building a system which decodes this sample into a sequence of words [2]. Thus, speech recognition is often also dubbed as speech-to-text conversion.



The text transcription decoded from a given speech sample using the Automatic Speech Recognition (ASR) system can either be used as is, i.e as a final output, or can be further processed using a variety of natural language processing units.

There are various parameters that characterize the efficiency of the Speech Recognition system, such as:

- (a) Whether the given speech sample is an isolated word or a continuous sequence of words
- (b) Whether the given speech sample was simply read by the speaker, or if it was spontaneous
- (c) Whether the system is speaker dependent or speaker independent
- (d) Whether the system is defined on a small vocabulary or a large vocabulary
- (e) Whether the Signal-to-Noise Ratio (SNR) is high or low
- (f) Whether or not a noise-reducing, high quality microphone was used for recording the speech

In the current systems, high performance is being observed on speaker-independent speech samples on a large vocabulary of words, spoken as read speech in a favourable environment; and on a moderate vocabulary of words spoken spontaneously over the phone. It has been observed that the interdisciplinary collaboration and the availability/affordability of high computing power has led to this improved performance over the years. [1]

1.1 Project Overview

The purpose of this project is to study the intricacies of the application, Automatic Speech Recognition, and to understand the various Deep Learning models used in a variety of ASR frameworks. Furthermore, the application is studied in the context of the Kaldi toolkit, wherein, various algorithms/methodologies used for the implementation of the speech recognition system are looked into. The performance of the speech recognition system shall be explored on a variety of these deep learning model frameworks using the Kaldi toolkit.

One of the primary purposes of this project is to study the effect of various acoustic modeling frameworks on the performance of the system, and to understand the reasons behind the improvement/deterioration in performance across different acoustic modeling frameworks.

1.2 Goals of the Project

The primary goal of this project is to explore the word error rates (WER) and sentence error rates (SER) obtained in the context of various datasets and deep learning model frameworks using the Kaldi toolkit. The improving performance over different deep learning model frameworks shall be tracked and the reasons behind the improvement in the performance of each framework shall be studied. By the end of this course, I intend to study the state-of-the-art frameworks that are being used today for the purpose of speech recognition and try to understand the reasons behind the improved performance of each framework.

Chapter 2

Automatic Speech Recognition

Automatic speech recognition can be understood as the conversion of a given speech input into corresponding text transcription. One of the better methods used widely today to model speech recognition systems is the probabilistic approach. A speech signal corresponds to a word or a sequence of words, as described in the dictionary. Thus, every word/sequence of words is scored based on the acoustic and linguistic properties, and the word sequence with the best score is given as the corresponding text transcription for that audio signal.

$$P(Y) = \max_w P(W|O)$$

Using Bayes Rule,

$$P(W|O) = \frac{P(O|W).P(W)}{P(O)}$$

Since $P(O)$ is independent of W , the MAP decoding can thus be represented as:

$$P(W|O) = \operatorname{argmax}_w P(O|W).P(W)$$

In speech recognition systems, the first parameter, i.e, $P(O|W)$ is represented by the acoustic model, and the second parameter, i.e, $P(W)$ is represented by the language model [3]. Thus, the process of speech recognition can be broadly divided into consecutive steps, as shown in Fig.1

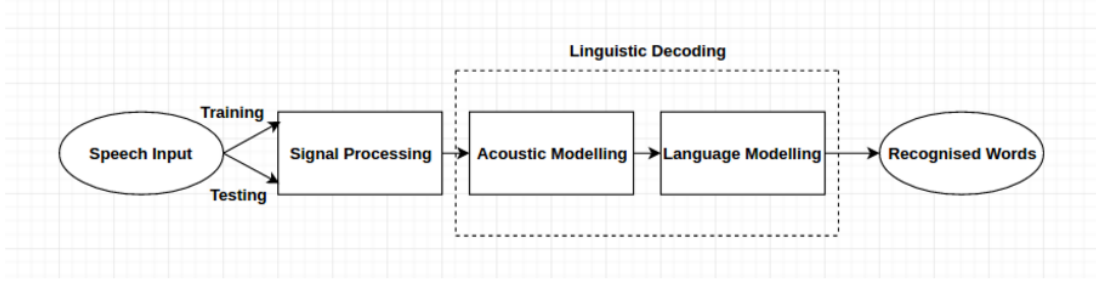


Fig.1 Outline of speech recognition system

2.1 Signal Pre-Processing and Feature Extraction

As the air comes out of the lungs, the twin infoldings of the vocal cords vibrate which results in the modulation of the flow of air. This air is further modulated by the vocal tract, velum, lips and tongue. This coarticulation leads to the production of various speech sounds/phonemes. These sounds/phonemes are dependent on the airstream, position of the glottis, position of the soft palate, position of the lips, active and passive articulators, etc. There are around 45 phoneme sounds in English, the number of which varies depending on the dialect of the region.

Over the years, through experiments it was concluded that the features motivated by speech perception work better than those motivated by speech production. As the sound enters the ear, it is converted into vibrations in the cochlear fluid. The cochlear fluid has several strands of little hairs which are sensitive to a range of frequencies, thus acting as band-pass filter. Using the human physiology as justification, a certain range of frequencies was computed which could be focused on. [2]

$$Mel\ Frequency = 2595 \log_{10} \left(1 + \frac{freq}{100} \right)$$

The Shannon Sampling Theorem states that a bandwidth limited signal can be reconstructed if the sampling frequency is more than double that of the maximum frequency. The speech recorded via microphones is usually discretized at a frequency of 16kHz. In the signal processing step, the end-point detection is done to remove the parts of the speech, added as a result of background noise, but resemble a phoneme model more than that of the silence

model. The features are extracted from these processed signals.

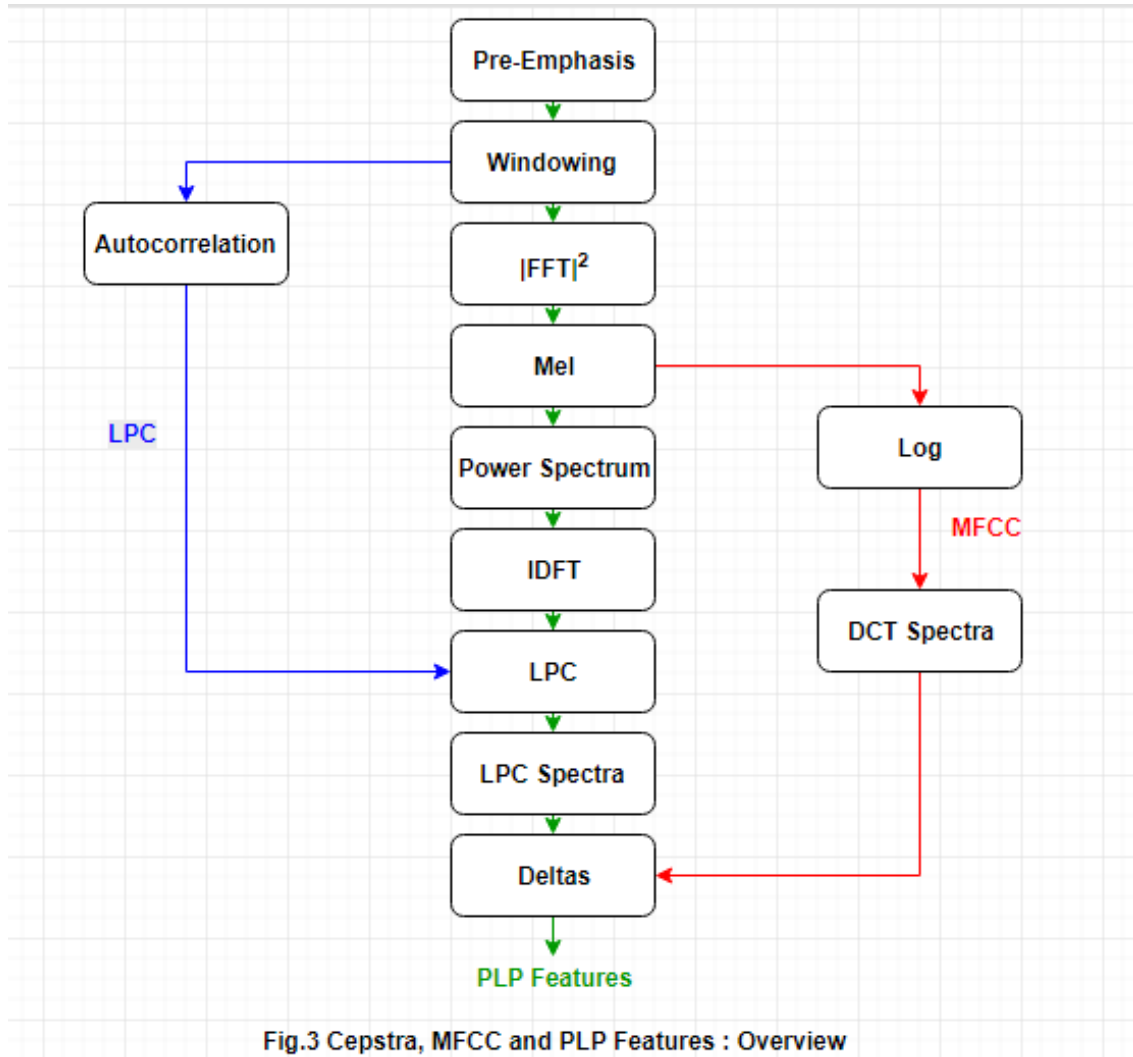


Fig.3 Cepstra, MFCC and PLP Features : Overview

The features of a speech signal are usually estimated over sequences with time interval of uniform length, and each overlapping with the next sequence. Such sequences are referred to as frames. Each frame typically has a length around 25ms and it overlaps with the next frame for a time interval of around 10ms. Fourier transformation is applied over every frame to translate it to the spectral domain, and features are thus estimated over each frame in the spectral domain. The main goal of feature extraction is to capture essential information that shall be useful in the identification of a particular word. In the 10ms time interval, the speech signal is almost periodic, for which the power spectrum is calculated. The cepstrum is the logarithm of the magnitude of the spectrum of a frame, which preserves the information of the vocal tract. The Mel Cepstrum is defined as the Discrete Cosine Transform (DCT) of filter

outputs $S(M)$:

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

Common feature extraction techniques include Mel Frequency Cepstrum Coefficient (MFCC), Filterbank Coefficients, Perceptual Linear Prediction (PLP) features, etc. [4]

2.2 Acoustic Modeling

A phoneme is the smallest sub-unit with a distinguishable meaning, based on which speech can be represented. Thus, words can be represented as a sequence of phonemes, and acoustic modeling comes down to pronunciation modeling. Acoustic modeling of speech, in general, refers to the process of estimating the statistical representations for the feature vectors, computed for a frame in a speech signal. This means that in acoustic modeling, for every frame, the feature vector is converted into probabilities of particular labels over some set of states, which is nothing but a classification problem. Acoustic modeling can be achieved through models such as Gaussian Mixture Models (Guassian distribution under each state), Hidden Markov Models, segmental models, super-segmental models, neural networks, etc [5].

Each state is represented by a senone (sound unit). The senone can be a monophone which is context-independent, or a n-phone (diphone, triphone, etc). The choice of the senone makes a huge difference in the speech recognition task. At the end of acoustic modeling, each feature vector representing a frame in the speech sample is labelled as a sequence of the states, constrained by the known word sequences in the lexicon. Thus, this process can also be understood as obtaining the alignments of the states.

2.2.1 GMM-HMM Acoustic Modeling framework

The Gaussian distribution is the most common probability distribution curve which is parameterised by the mean and variance of the data points, with the location dependent on the mean of the data points and the spread dependent on the variance of the data points. Gaussian Mixture Models (GMM) are a mixture of different Gaussian components, characteristic to their

respective mean and variance.

$$p(x) = \sum_{j=1}^P P(j).p(x|j)$$

$$p(x) = \sum_{j=1}^P P(j).N_j(x; \mu_j, \sigma^2)$$

Hidden Markov models track the temporal variability in a signal. In the Hidden Markov Models for ASR, it is assumed that the acoustic observation produced by a state is conditionally independent of all other acoustic observations. In GMM-HMM acoustic modeling, the acoustic observation in a state is assumed to be strongly Gaussian, and the family of components of the GMM can model any distribution. Thus, this model creates a sequence of Gaussian models for each of the states, and the acoustic observations associated with them. [3]

2.2.2 DNN-HMM Acoustic Modeling framework

A DNN is a feedforward neural network in which the connections exist between the nodes/neurons in one layer to those in the next layer. A DNN consists of more number of hidden layers. Deep Neural Networks are difficult to train because of the large number of parameters that need to be estimated due to the high number of neurons in the hidden layers. Deep Neural Networks are supposed to be better classifiers than Gaussian Mixture Models, and are thus used for acoustic modeling.

ASR systems trained with the DNN-HMM acoustic modeling frameworks are shown to work more accurately as compared to the systems which incorporate the GMM-HMM acoustic modeling framework. The DNN-HMM acoustic modeling framework represents a hybrid system. The DNN models the posterior probability of a state given an acoustic observation. The DNN handles time-bound sequences poorly, but has a good discriminative power and can incorporate contextual information of a state rather easily. [4]

2.2.3 TDNN-HMM Acoustic Modeling framework

Acoustic models than can model the long-term dependencies between acoustic events are necessary to capture the temporal properties of speech. Recurrent neural networks are apt for this purpose, but the only drawback with them is that parallelisation is not possible due to the recurrent connections. This is possible in feedforward neural networks, and thus, time-delay neural network is the better alternative.

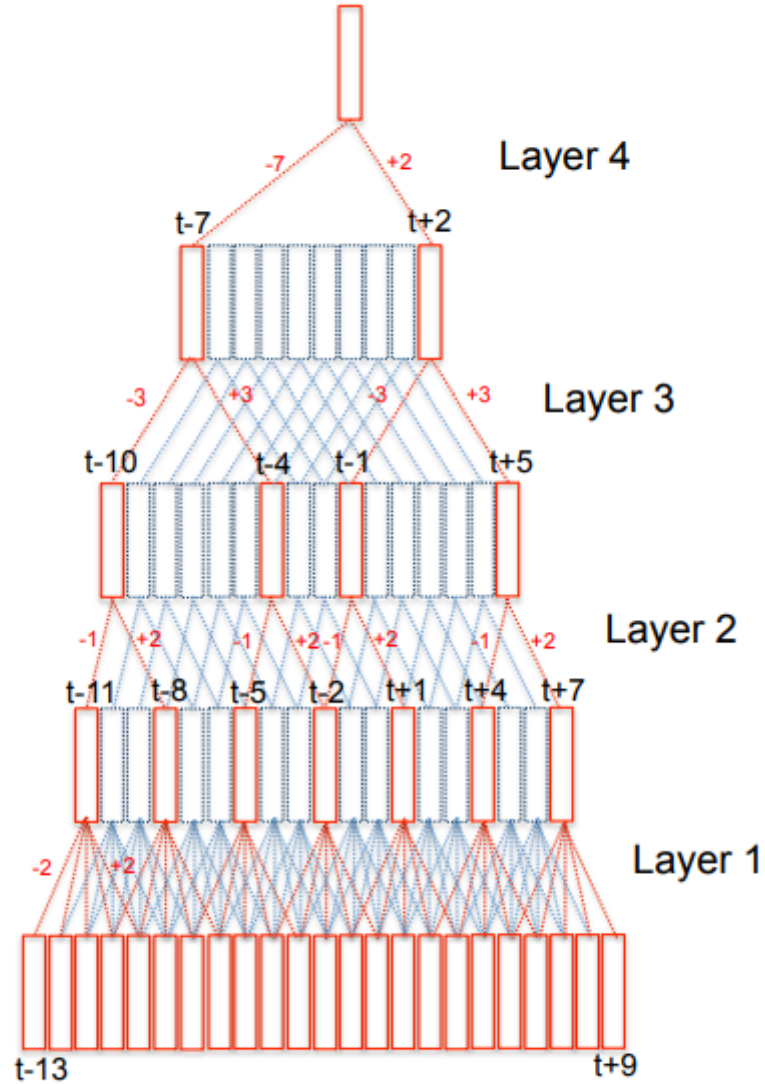


Fig.4 Computation in TDNN with (red) and without (blue) sub-sampling

As shown in Figure 4[5], the architecture of TDNN is such that a frame in one layer is dependent on two frames in the next layer, as defined by the left context and the right context.

Thus, to all the frames that have been spliced in the context (marked in red), there is a left context and a right context in the next layer that it is dependent on.

2.2.4 SGMM-HMM Acoustic Modeling framework

As discussed in 2.2.1, the GMM-HMM acoustic modeling framework trains a number of parameters for each component of the mixture used to model the distribution. In the Subspace Gaussian Mixture Model framework, the super-vector which is common to all the component Gaussians in the mixture is constrained such that it is common to all the states. This constraint is justified because each state has a high degree of correlation with the other, as they are so designed to incorporate the context dependency. [6]

Thus, the distribution of states is represented by a relatively lower dimensional vector (as compared to the super-vector) which incorporates the various coordinates of the subspace. This compact representation of the HMM-state distributions results in the estimation of a fewer number of parameters in the Gaussian distribution. The SGMM-HMM acoustic modeling framework can also be extended to multilingual experiments, or for languages with limited resources.

2.2.5 TDNN-LSTM Acoustic Modeling framework

The knowledge of future context information helps in acoustic modeling. The context can be obtained in feed forward neural networks by appending a fixed set of future frames to the input acoustic observation, or by convolving the input acoustic observation with the time of the future context. To model the future context in the acoustic modeling using LSTMs, TDNNs can be used. TDNNs can be used for temporally convolving with LSTMs by stacking the TDNNs over the LSTM units, by stacking the LSTM units over TDNNs, and by interleaving TDNNs and LSTM units. [7]

The latency of the model can be affected by changing the input context, chunk width, output delay, etc. Using additional temporal context improves the performance of the model for ASR systems with TDNN-LSTM acoustic models. Interleaving the temporal convolution

of LSTM layers is effective for modeling the future temporal context.

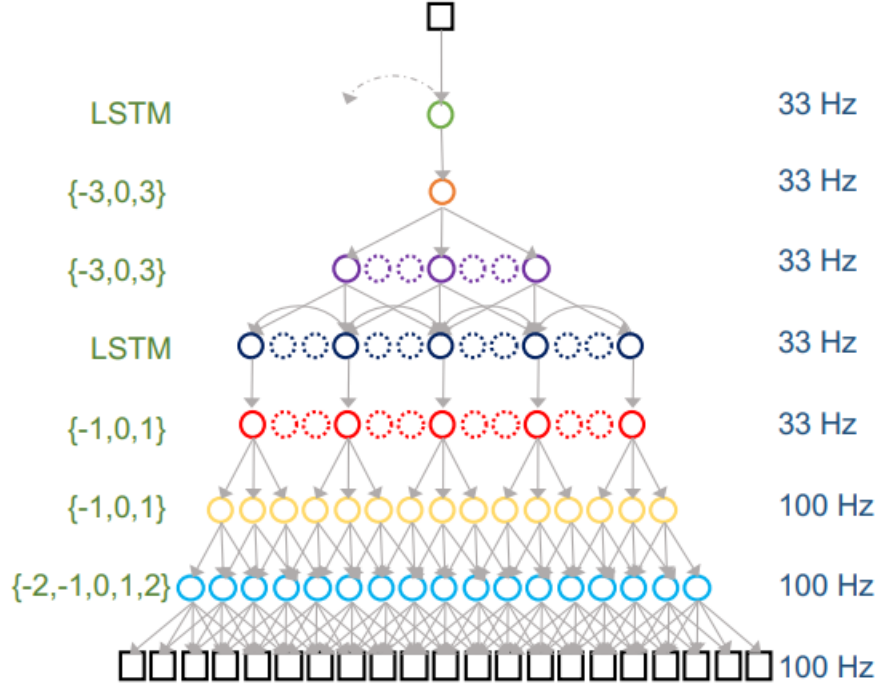


Fig. 5 Dependencies among activations in a stacked TDNN-LSTM network with interleaved temporal convolutions. The convolution kernel input contexts are on left and the layer-wise frame rates are on the right

2.2.6 RNN-LSTM Modeling framework

LSTM units are a type of recurrent neural networks which contain memory blocks in the recurrent hidden layer. These memory blocks contain memory cells with self-connections, which store the temporal information. They also have multiplicative units called gates which control the flow of information in the network. The LSTMs compute the probability of likelihood of the acoustic data in speech recognition. In a recurrent neural network, the distribution for each frame is different, and thus, there is no need for modeling of three different distributions for individual phonemes. [8]

LSTM models the acoustic trajectory of each phone, and it is a good representation of this trajectory. The second layer in the LSTM layer's state represents the final frame of the phoneme. The connectionist temporal classification (CTC) with LSTM's acoustic modeling

shows a higher accuracy eliminating the need for context-dependent modeling. [9]

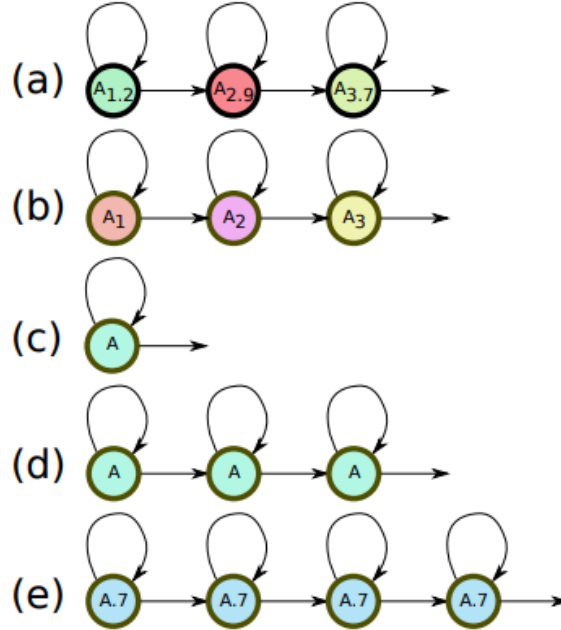


Fig.6 Simple left-to-right HMM topologies: (a) A conventional 3-state context-dependent HMM; (b) A 3-state context-independent HMM; (c) A one-state phone HMM; (d) A tied-state context-independent phone model with minimum duration of 3 states; (e) A tied-state context-dependent phone model with minimum duration of 4 states.

2.3 Language Modeling

Language modeling is the assignment of probabilities $PL(W)$ to sequences of words that form valid sentences in the language. Such language models are trained on the generic text sequences in the dataset on which recognition is being performed [6]. In language modeling, a deterministic grammar is formulated from the text (sequence of words) of sentences spoken in the dataset, based on which, acceptable sentences/sequence of words are determined. Thus, the probability of the occurrence of a word string is 1, if the sentence is acceptable, and 0 otherwise. Stochastic language models, such as n-gram models can also be estimated, which define the probability of occurrence of an ordered sequence of a group of words. Thus, based on the statistical language model, the probability of a word sequence can be modelled as:

$$P(W) = P(w_1, w_2, w_3, \dots, w_n)$$

$$P(W) = P(w_1).P(w_2|w_1).P(w_3|w_1, w_2)...P(w_n|w_1, w_2, w_3, \dots, w_{n-1})$$

However, it was observed that predicting the probability of a word with the input history of all the previous words was impossible, and thus, in practice, the word relationship is estimated only over a span of N-words, or N-1 previous words [7].

2.4 Linguistic Decoding

The decoding process seeks to find the word sequence that best fits the acoustic observations of a particular speech sample. The best fit can be obtained by maximising the posterior probability (MAP) as obtained in:

$$P(W|O) = \operatorname{argmax}_w P(O|W).P(W)$$

The parameters for the acoustic modeling and language modeling are given scores, based on which the score is computed for this posterior probability. Thus, the decoding process can be construed as searching for the best possible word sequence with the highest posterior probability and score, given the acoustic observations. To obtain the best score for the word sequence, the acoustic and language models should be balanced. The triphone modeling of the states in acoustic modeling leads to higher probabilities of the acoustic model.

In systems with large vocabularies, searching all possible word sequences is not feasible. Also, in connected word recognition, the number of words in the speech utterance, and the word boundaries are not known. While Viterbi Decoding performs an exact search in the word space in an efficient manner, it's not feasible for large vocabulary tasks. More efficient algorithms such as Beam Search, Multipass Search (Two-Stage Decoding), Best-First Search (A* Search), Weighted Finite State Transducers can be used.

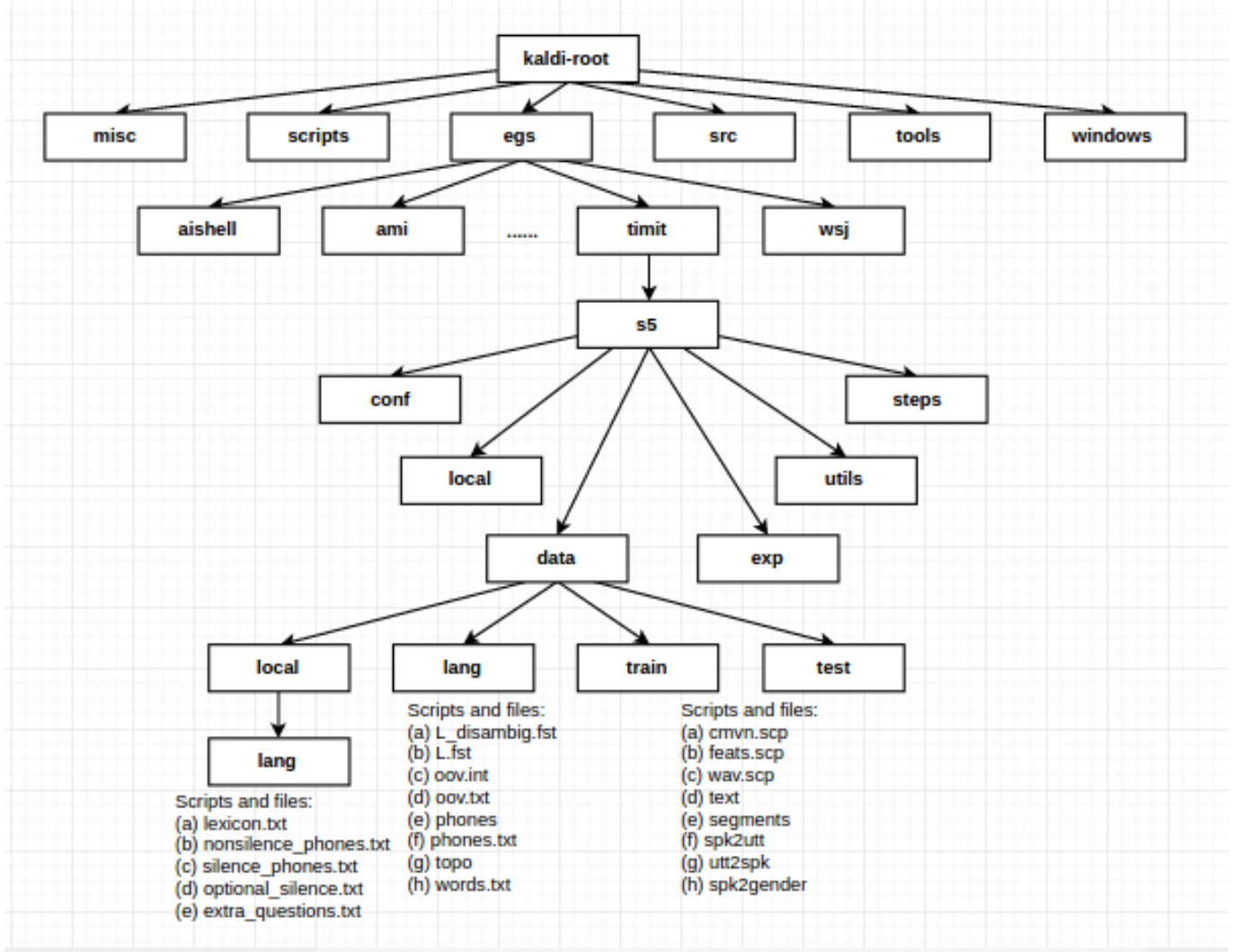
Chapter 3

Automatic Speech Recognition using Kaldi

Kaldi toolkit is a free, open-source toolkit developed for speech recognition research. It is a Finite State Transducer (FST) based framework (compiled against the OpenFST toolkit), providing extensive linear algebra support through BLAS and LAPACK routines. Kaldi contains recipes for the widely-used speech databases in the speech-recognition community such as those provided by the Linguistic Data Consortium (LDC) [8].

3.1 Structure of the Kaldi Toolkit

The Kaldi toolkit comprises of the following main directories: `egs`, `misc`, `src`, `tools` and `windows`. The `egs` directory contains various folders with recipes in a variety of frameworks, for various datasets such as `aishell`, `ami`, `an4`, `timit`, `wsj`, `rm`, etc. The `misc` directory contains the source and pdf for some Kaldi-related papers, a few HTK-Kaldi conversion scripts, etc. The `src` directory contains various scripts and binary files that shall be used in the recipes throughout the toolkit. The `tools` directory contains various toolkits that Kaldi uses for the compilation of the recipes.



We shall primarily be using the egs directory for the implementation of the recipes in Kaldi. Over the years, there has been an upgrade in the versions of recipes for the datasets in Kaldi, and s5 is the latest version being used. The recipes directory for a particular dataset typically consists of the conf, local, data, exp, utils and steps sub-directories. The conf sub-directory consists of the basic configurations required for the implementation of the recipes, such as that for the mfcc, fbank, etc. The exp sub-directory is the output directory, where the alignments, scores after decoding for various recipes of the dataset, etc. shall be stored. The local sub-directory consists of the scripts pertaining to that particular dataset, such as that for data preparation, language modelling, etc. The utils and steps sub-directories are common to all the datasets and contain various scripts that the recipes shall be using through the implementations.

The data sub-directory consists of the local, lang, train, test sub-directories. The lang sub-directory in the local sub-directory is concerned with the language modelling of the system, and contains files such as lexicon.txt (list of words expressed as a sequence of phonemes), nonsilence_phones.txt (list of the non-silent phonemes), silence_phones.txt (list of silent phonemes), optional_silence.txt (list of unknown phonemes). The lang sub-directory in the data directory consists of the FSTs and phone-lists created from the local sub-directory. The train and test sub-directories consists of various files such as text (utterance ids and corresponding transcriptions), wav.scp (utterance ids and corresponding locations of the audio files), utt2spk (utterance ids and corresponding speaker ids), etc.

3.2 Signal Processing and Feature Extraction in Kaldi

In this stage, features distinct to speech are extracted from the input speech signal. In Kaldi, features such as MFCC, Filterbank, PLP, etc can be computed. The Kaldi toolkit also supports a variety of feature transforms, projections and other feature operations such as:

- (a) Linear Discriminant Analysis (LDA)
- (b) Heteroscedastic Linear Discriminant Analysis (HLDA)
- (c) Maximum Likelihood Linear Transform (MLLT) estimation
- (d) Exponential Transform (ET)
- (e) Cepstral Mean and Variance Normalisation (CMVN) [9]

Kaldi usually loads the inputs into its files and again stores results in the file. It has a list of executables in the directory for every speech recognition task. The list of executables for signal analysis and feature extraction in Kaldi are:

- (a) apply-mfcc
- (b) compute-mfcc-feats
- (c) compute-plp-feats
- (d) add-deltas
- (e) compute-cmvn-stats, etc.

3.3 Acoustic Modeling in Kaldi

In this step, the HMM-parameters for the senomes (monophones, diphones, triphones, etc.) are computed using the maximum likelihood estimation criterion, which estimates the parameters in regard with the model. In Kaldi, the Acoustic Models are trained hierarchically, i.e, the higher model trains from a previously trained lower model. First, a monophone acoustic model is trained using the MFCC features, Δ -features and $\Delta\Delta$ -features extracted for the speech sample. Then, the triphone acoustic model (tri1) is trained on this monophone acoustic model. In the next branch of experiments, the tri1 acoustic model uses LDA+MLLT to train. This is referred to as the tri2 acoustic model. In another branch of experiments, SAT is applied on the tri2 acoustic model, which is referred to as the tri3 acoustic model.

In Kaldi, the diagonal and full covariance structures for Gaussian Mixture Models are supported. The acoustic model class *AmGMM* represents a collection of *DiagGmm* objects, indexed by pdf-id's that correspond to context-dependent HMM states. For the Subspace Gaussian Mixture Models, there is one single class *AmSgmm* which represents the whole collection of pdf's.

3.4 Language Modeling in Kaldi

The main aim of the language modeling step is the creation of the G.fst, which is a weighted finite state transducer of the grammar. In order to create G.fst, the files that are required are:

- (a) extra_question.txt
- (b) lexicon.txt
- (c) nonsilence_phones.txt
- (d) optional_silence.txt
- (e) silence_phones.txt

From the above-mentioned files, several Symbol-Table files are created in the OpenFST format which are used by Kaldi to map the integer and text forms of the symbols. Also, the list of phones, out-of-vocabulary characters, etc. are extracted. The following is the list of

files extracted from the above-mentioned files:

- (a) phones.txt
- (b) words.txt
- (c) oov.int
- (d) topo
- (e) silence.txt
- (f) nonsilence.txt, etc.

Furthermore, *arpa2fst* script in Kaldi is used to convert this ARPA-format language model into a weighted finite state transducer. SRILM toolkit is used in Kaldi for the construction of language models.

3.5 Decoding in Kaldi

Using a combination of the Acoustic Model probabilities and the Language Model probabilities, the word string is estimated. Speech Recognition is a combination of the pattern recognition technique and the global search problem. The concept of N-best lists is used in the decoding step in Kaldi. From the N-best lists, the lattices of the speech sample are extracted [10]. The decoding is done on the decoding graph HCLG, which is as constructed:

$$HCLG = H \circ C \circ L \circ G$$

Here,

G is an acceptor which encoded the Language Model

L is the lexicon

C represents the context dependency

H contains the HMM definitions, output symbols of which represent the C-D phones

The composition of H, C, L and G results in a transducer that maps the given word sequence to a sequence of HMM states. However, in Kaldi, it is important to note as to in which order the composition is to take place, i.e, first determinisation, then minimisation.[10]

The list of executables for speech decoding in Kaldi are:

- (a) gmm-latgen-faster
- (b) gmm-latgen-faster-parallel
- (c) gmm-latgen-biglm-faster, etc.

Chapter 4

Experimentation

4.1 Data

The experimentation has been done on the TIMIT Dataset provided by the Language Data Consortium (LDC). This dataset has a total of 630 speakers, and each speaker has 10 utterances, i.e, it has a total of 6300 utterances. The speakers have been taken from 8 major dialect regions in the United States of America, labelled dr1 - dr8. The total male to female ratio among the speakers is 70:30. The 6300 utterances can be broken down as follows:

- (a) 2 dialect sentences designed at SRI International (denoted by SA)
- (b) 450 phonetically compact sentences designed at MIT (denoted by SX)
- (c) 1890 diverse sentences designed at Texas Instruments (denoted by SI)

SA were spoken by all the 630 speakers. In SX, each speaker read 5 of these sentences and each text was spoken 7 times. In SI, each speaker read 3 of these sentences and each text was spoken only once. The dictionary of the TIMIT dataset has a total of 6229 entries/words.

4.2 Training

The training for the TIMIT dataset was first done on the GMM-HMM modeling framework, in which the states were represented by a monophone HMM. Based on the alignments achieved in this framework, the acoustic modeling was done to build a system in which the

states were represented by triphone HMMs. Various feature dimensionality reduction and discriminative training methods were applied to obtain three versions of the triphones, and the speech recognition system was tested for each of these frameworks.

Based on the alignments of the triphones achieved in the acoustic modelling, by applying a combination of Linear Discriminant Analysis (LDA), Speaker Adaptive Training (SAT) and Maximum Likelihood Linear Transform (MLLT), commonly referred to as tri3, the acoustic model for a DNN-HMM modeling framework was built and the accuracy for this system was estimated. Furthermore, the TDNN-HMM modeling framework was tested upon as well, by training the acoustic model with a TDNN, alignments of which were obtained from the tri3 triphones.

The training is also done on the SGMM-HMM model wherein the acoustic modeling is done by the sub-space GMM (SGMM) on the tri3 alignments. Furthermore, acoustic modeling is also done on the system discriminatively trained by the boosted MMI technique on the SGMM-HMM modeling framework. Furthermore, Recurrent Neural Networks (RNN) and Time-Delay Neural Networks (TDNN) were combined with Long Short-Term Memory units for acoustic modeling in speech recognition. The Word Error Rates (WER) were estimated on these modeling frameworks in speech recognition using Kaldi.

4.3 Results and Discussion

The KALDI system produces lattices as recognition result. To obtain the best path, the standard KALDI procedure was followed and the best WER was reported, based on evaluation on a set of language model scaling factors [11]. In a monophone system, the ASR system is looking to match the audio to a particular phoneme sound. However in the triphone system, the ASR system is looking to match the audio to a sequence of phonemes (in this case, sequence of three phonemes). Also, in the monophone system, we are looking only for one particular instance of the phoneme, whereas in the triphone system, multiple HMM instances are created for a particular phoneme with different transitions of the states in the system. Thus, the latter system takes into account the context of the phoneme in the audio sample,

thus resulting in the improvement of the accuracy.

Model	Word Error Rate
GMM-HMM (Monophone)	34.2%
GMM-HMM (tri1: Deltas + Delta-Deltas)	25.9%
GMM-HMM (tri2: LDA + MLLT)	24%
GMM-HMM (tri3: LDA + MLLT + SAT)	21.4%
DNN-HMM (tri3)	22.6%
TDNN-HMM	22.1%
SGMM2	19.8%
MMI + SGMM2	19.9%
TDNN-LSTM	18.1%
RNN-LSTM	17.4%

For automatic speech recognition purpose, feature vectors are estimated which are usually 13 dimensions per frame. The delta + delta-delta measures are expected to contribute to the speaker information in the features. In this method, 13 dimensions each are augmented to the feature vector corresponding to delta and delta-deltas respectively. Delta measure refers to the rate of change of the mel-cepstral coefficients over a frame, and delta-delta measure refers to the acceleration of the cepstral coefficients [12]. In theory, it is sufficient if we have two consecutive frames to compute the delta measure and three consecutive frames to compute the delta-delta measure. However, in practice, 5 consecutive frames are used to compute each of these measures. The addition of this speaker information is expected to give an improvement in the accuracy of this system with respect to the triphones.

In the feature vectors, 13 dimensions are estimated per frame. In this method, the left-context and right-context of 3 frames each are taken into account and the feature vector is

expanded by splicing these 7 frames, thus increasing the dimensionality of the feature vector to that of (13×7) 91 dimensions. This is done to include the speaker information in the features. Further, the dimensionality of this vector is reduced to 40 using the LDA (Linear Discriminant Analysis) and further decorrelation is done using MLLT. Thus, this method ensures the feature vector has more speaker information and the triphones based on these features produce better results.

The DNNs are better classifiers than GMMs, as they can generalize better with a smaller number of parameters, and also, DNNs enable including context of the frames for the acoustic modelling. For these reasons, when DNN is used as the acoustic model on the tri3-modified triphones, it performs better than the GMM acoustic models.

A TDNN unit has the ability to relate and compare current input to the past history of events. Each TDNN unit has the ability to encode temporal relationships within the range of N delays. Higher layers can attend to larger time spans, so local short duration features at the lower layer and more complex longer duration features at the higher layer. The learning procedure ensures that each of the units in each layer has its weights adjusted in a way that improves the networks overall performance. Thus, the TDNN acoustic model performs better than the DNN acoustic model [13].

Using additional temporal context resulted in an improvement in the performance of the TDNN-LSTM acoustic modeling framework. In the experiment performed, a TDNN was used between two successive LSTM units. Bidirectional long short-term memory (BLSTM) acoustic models provide a significant word error rate reduction compared to their unidirectional counterpart, as they model both the past and future temporal contexts. The increase in the context in this framework results in a better classification and thus an improved performance than that of the other models using HMMs. [11]

Recurrent Neural Networks exploit a dynamically changing contextual window in the input layer, as compared to a static input layer size in the feed-forward networks. In the Long Short-Term Memory architecture, the drawbacks of RNNs are taken care of, because of the ability to store more contextual information, thus improving the performance of the RNN-LSTM acoustic modeling framework in speech recognition. Another cascade operation that can be tested upon to improve the performance of this framework is to use dropout on the different RNN units.

Chapter 5

Conclusions and Future Work

In this report, I have shown the performance of the speech recognition system for various acoustic modeling frameworks using the Kaldi toolkit. The performance of GMM-HMM, DNN-HMM, RNN-HMM, TDNN-HMM, TDNN-LSTM, RNN-LSTM acoustic modeling frameworks in Automatic Speech Recognition were tested upon. Research work in speech recognition is heading towards end-to-end speech recognition systems. There has been an improvement in performance of Speech Recognition using the Long Short-Term Memory (LSTM) units and Bidirectional Long Short-Term Memory (BLSTM) units. I intend to study the Generative Adversarial Network (GANs) in the future and test their performance in speech recognition as an addition to my current work.

Bibliography

- [1] A Brief Introduction to Automatic Speech Recognition Jim Glass (glass@mit.edu) MIT Computer Science and Artificial Intelligence Laboratory November 13, 2007
- [2] Lecture 1 Signal Processing and Dynamic Time Warping Michael Picheny, Bhuvana Ramabhadran, Stanley F. Chen, Markus Nussbaum-Thom Watson Group IBM T.J. Watson Research Center Yorktown Heights, New York, USA
picheny,bhuvana,stanchen,nussbaum@us.ibm.com
- [3] Lecture 2 Signal Processing and Dynamic Time Warping Michael Picheny, Bhuvana Ramabhadran, Stanley F. Chen, Markus Nussbaum-Thom Watson Group IBM T.J. Watson Research Center Yorktown Heights, New York, USA
picheny,bhuvana,stanchen,nussbaum@us.ibm.com
- [4] Phonetic Classification in TensorFlow; Timo van Nidek S4326164 First supervisor/assessor: Prof. Dr. Tom Heskes t.heskes@science.ru.nl
- [5] A time delay neural network architecture for efficient modeling of long temporal contexts Vijayaditya Peditinti¹, Daniel Povey^{1,2}, Sanjeev Khudanpur^{1,2} ¹Center for Language and Speech Processing ²Human Language Technology Center of Excellence Johns Hopkins University, Baltimore, MD 21218, USA
- [6] MULTILINGUAL ACOUSTIC MODELING FOR SPEECH RECOGNITION BASED ON SUBSPACE GAUSSIAN MIXTURE MODELS Lukas Burget¹, Petr Schwarz¹, Mohit Agarwal², Pinar Akyazi³, Kai Feng⁴, Arnab Ghoshal⁵, Ondrej Glembek¹, Nandendra Goel⁶, Martin Karafiat¹, Daniel Povey⁷, Ariya Rastrow⁸, Richard C. Rose⁹, Samuel Thomas⁸ ¹ Brno University of Technology, Czech Republic, burget,schwarzp@fit.vutbr.cz;

2 IIIT Allahabad, India; 3 Bogazici University, Turkey; 4 HKUST, Hong Kong; 5 Saarland University, Germany; 6 Virginia, USA; 7 Microsoft Research, Redmond, WA; 8 Johns Hopkins University, MD; 9 McGill University, Canada

- [7] Low latency acoustic modeling using temporal convolution and LSTMs Vijayaditya Peditinti , Yiming Wang, Daniel Povey, Sanjeev Khudanpur
- [8] Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling Hasim Sak, Andrew Senior, Francoise Beaufay
- [9] CONTEXT DEPENDENT PHONE MODELS FOR LSTM RNN ACOUSTIC MODELING Andrew Senior, Hasim Sak, Izhak Shafran Google Inc., New York
andrewsenior,hasim,izhak@google.com
- [10] Decoding and WFSTs Steve Renals Automatic Speech Recognition ASR Lecture 13 9 March 2017
Mohri (2008) Mohri, Pereira, and Riley (2008). Speech recognition with weighted finite-state transducers. In Springer Handbook of Speech Processing, pp. 559-584. Springer, 2008. <http://www.cs.nyu.edu/~mohri/pub/hbka.pdf> Decoding and WFSTs in Kaldi <http://danielpovey.com/files/Lecture4.pdf>
- [11] Low Latency Acoustic Modeling Using Temporal Convolution and LSTMs
Vijayaditya Peditinti