VERY DEEP CNN ACOUSTIC MODELLING FROM SPEECH WAVEFORM

Atul Agarwal

Master of Technology Thesis

June 2018

International Institute of Information Technology, Bangalore

VERY DEEP CNN ACOUSTIC MODELLING FROM SPEECH WAVEFORM

Submitted to International Institute of Information Technology,
Bangalore
in Partial Fulfillment of
the Requirements for the Award of
Master of Technology

by

Atul Agarwal
MT2016005

International Institute of Information Technology, Bangalore
June 2018

Dedicated to

My Family

Thesis Certificate

This is to certify that the thesis titled **Very Deep CNN acoustic modelling from speech waveform** submitted to the International Institute of Information Technology, Bangalore, for the award of the degree of **Master of Technology** is a bona fide record of the research work done by **Atul Agarwal**, **MT2016005**, under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. V. Ramasubramanian

Bengaluru,
The 5th of June, 2018.

VERY DEEP CNN ACOUSTIC MODELLING FROM SPEECH WAVEFORM

Abstract

Conventionally Convolutional Neural Networks (CNNs) for automatic speech recognition are used with Mel frequency cepstral coefficient (MFCCs) features or spectrogram. Recent end to end architectures uses connectionist temporal classification (CTC) loss with Recurrent neural network (RNN) and spectrogram/MFCC features which makes unsegmented labelling possible and makes it feasible to train in 'end to end' manner. However, recently CNNs have been used on raw waveform speech data for representation/feature learning as a part of front end and then the usual RNN/Long short term memory (LSTM) networks are used. RNNs along with LSTMs are computationally expensive, so there are studies which combine CNN with spectrogram/MFCCs along with CTC to get the results comparable to state of the art. Inspired from this and another work on very deep CNN for raw waveform on environmental sound we try to combine very deep CNNs for raw-waveform and CTC. We except various types of normalization of the raw waveform (with respect to duration, gain, pitch, channel, speaker etc.) prior to presentation to the representation learning network would help the system to gain in terms of accuracy or architecture complexity, by directly reducing the intra-class variability that the network has to cope with in classification. We also compare results of various length inputs to the proposed network. We start with frame level classification and then do the continuous phoneme decoding using CTC. We tune all the hyper-parameters while doing the frame level classification and use the same architecture then with CTC loss.

Acknowledgements

I would like to thank my supervisor, Dr. V. Ramasubramanian for his support and guidance throughout this work. I'd like to thank my colleagues in speech group. Finally I'd like to thank my friends and family for supporting me in difficult times.

Contents

List of Figures

List of Tables

List of Abbreviations

ASR . . . . . Automatic Speech Recognition

CNN . . . . . Convolution Neural Network

CDBN . . . . . Convolutional Deep Belief Network

CRBM . . . . . Convolutional Restricted Boltzmann Machine

CTC . . . . . Connectionist Temporal Classification

DBN . . . . . Deep Belief Network

DNN . . . . . Deep Neural Network

IIITB . . . . . International Institute of Information Technology Bangalore

LSTM . . . . . Long Short Term Memory

MFCC . . . . . Mel Frequency Cepstral Coefficient

RBM . . . . . Restricted Boltzmann Machine

ReLU . . . . . Rectified Linear Unit

RNN . . . . . Recurrent Neural Network

CHAPTER 1

INTRODUCTION

Speech signals are one of the complex signals known to humans. This complexity is due to various components like accent, pronunciation, articulation, pitch, noise, etc. Despite being so complex, humans are able to overcome all these components effortlessly but it becomes difficult for Automatic Speech Recognition (ASR) systems to handle all these components. For example, there may be some information related to accent which may be helpful to system but is not available and hence task becomes more tedious. Extracting a representation which can deal with all the complexity in the signal and environment is still an open problem.

A very commonly used representation for speech signal is Mel Frequency Cepstral Coefficient (MFCC).This representation is inspired from the human auditory systems. From traditional ASR [1] to recent DeepSpeech [2] systems these representations are used for speech signals Though MFCCs have given good results for speech recognition task, they throw away a lot of information which can help to deal with some problems mentioned above. We believe that using raw wave-form with very deep Convolution Neural Network (CNN) can help us to eliminate the intermediate representation like MFCCs and get comparative results.

CNN architecture is the biological modification of the multi-layer perceptron proposed in [3]. Deep CNNs are powerful representation learning tool which was first demonstrated in [4] and then with further improvements over the years [5–7] deep CNNs were able to surpass the human accuracy in object recognition. While these

developments were done for vision tasks, the shallow versions of the CNNs with some other modifications were used for speech recognition task. These modifications were use of fully-connected layer, recurrent layer, long-short term memory (LSTM) units, etc [2]. The main limitation of these modification is that they were computationally expensive. The architectures which are being designed to do speech recognition are getting more and more computationally expensive due to use of LSTM units and recurrent layers. Also these architectures need huge amount of data to learn from or it over fits the data due to memory components. All the above computationally expensive components are trained for modeling the temporal behavior of the speech signal. 1D CNNs are also used for modeling these kind of temporal behavior. We believe that using 1D CNNs for both learning representation and modeling the temporal behavior of the speech signal will reduce the computational cost of the systems.

One of the major advances in sequence modeling task is Connectionist Temporal Classification (CTC) loss [8]. This loss has made it possible to label unsegmented sequences, which is needed for any speech system and have made it possible to train the system in an end-to-end manner. Also systems with this loss have outperformed the systems with the traditional setup. So we have combined this loss with 1D CNNs which help us to train the system in end-to-end manner without any preprocessing.

1.1 Aim and Objective

The main objective of this work is to study feasibility of 1D CNNs for representation learning and temporal modeling without any pre-processing of raw speech waveform. We also try some normalization techniques to reduce the variability that is dealt by the network. We have used TIMIT [9] corpus for all our experiments. We also perform various experiments with CNNs with different number of layers and different input dimension to exploit the representation power of CNNs.

1.2 Contributions

- We use only convolutional layers to exploit the representation learning power of the system.
- Use of only convolutional layers with CTC for raw waveform
- Applied normalization techniques and studied their effects on raw waveform.

1.3 Thesis Outline

Chapter 2 is the literature survey of all the representation learning schemes done for speech signals. Chapter 3 describes the architecture of the models used with all the variations. It also provides the necessary building blocks for understanding this thesis. Chapter 4 has the results of all the experiment done. Chapter 5 discusses the limitations and future work. Chapter 6 has the concluding remarks for this thesis work

CHAPTER 2

LITERATURE SURVEY

Traditional automatic speech recognition (ASR) systems have divided the speech recognition task in two fold.

- Feature extraction i.e. representation of speech signal for modeling (MFCC).
- Statistical modelling.
- Decoding the modelled sequence.

The MFCC have been used for a long time, and there has been very less development for alternate representation of speech signal until 2006. After the breakthrough of generative greedy layer wise pre-training of Deep Belief Network (DBN) [10] in 2006, researchers have started to search for alternate representation using neural architecture. One of the early works using deep belief network was done in [11, 12]. In this approach they used the conventional MFCC feature with context i.e. using feature vectors of neighboring frames to provide context to the system. In this system they pre-trained the DBN using Restricted Boltzmann Machine (RBM). The basic idea of RBM is it uses an energy function to model the relation between visible and hidden layer units. More the energy less stable is the model and less the energy more stable is the model. This was one of the earliest works in representation learning in which they got an error of 23% on core TIMIT dataset. Many other developments took place around RBM but the major success was Mean-Covariance RBM [13] which improved the error to 20.5% on the same timit data-set. Meanwhile, there were also experiments to learn representation

from raw speech signal [14]. This work feeds raw waveform to RBM and achieves a comparable result of 21.8%. The recent work on RBM includes DBN for i-Vectors based speaker recognition [15]. One of the major drawback of this approach was that its weights were tied to specific position but the phonemes are continuous in nature and can occur at any position which may degrade the performance of the system.

As the RBM were gaining popularity, they were facing scalability issues due to large weight space for inputs like images. To overcome this, Convolution Restricted Boltzmann Machine (CRBM) were proposed in for images [16]. CRBM were variant of RBM in which weights were shared with respect to the spatial structure of images or the other grid like topology. Inspired by this advancement there were studies for getting alternate representation of speech using CRBM. As CRBM was developed for images, researchers have used time frequency representation of the speech signal with the spectrogram as input [17]. They used convolutional deep belief network for various audio classifications like speaker identification, speaker gender classification and phone classification. They also used Convolutional-DBNs (CDBNs) for music tasks like genre classification task and artist classification. Their study showed that using representation which is generated using CRBM and spectrogram gave equivalent performances as that of MFCCs, and even surpass it in some tasks. In this work, they also demonstrated that CRBM was able to learn phoneme specific representation in an unsupervised manner via visualization of various phoneme classes. They also demonstrated visualization for speaker gender classification task which showed high energy bands for male speech and low for female. This was one of the earliest work which used spectrogram.

RBM and its variants were not the only methods to learn representation in unsupervised manner. Another approach for representation learning which gained popularity during the same period was use of auto-encoders [18]. Auto-encoders are the neural networks which tries to copy the input to its output with least distortion. An auto-encoder has two parts: encoder and decoder. Encoder projects the input data into latent space and decoder project data from latent space back to original input space. One major ad-

vantage of auto-encoder is that we can train multiple layers in one shot. Auto-encoders were mostly used for dimensionality reduction and feature extraction [19]. There are variants of auto-encoders which were used for representation learning. Majorly auto-encoders were used for the purpose of speech enhancement and de-noising [20,21]. The de-noising auto-encoders were trained by adding noise to data and trying to re-generate clean version of data from the latent space [22]. Initially the search for alternate representation started with using MFCC features as the input to the shallow auto-encoder in [23]. They were able to show that these kind of features were able to surpass the performance of the plain MFCC feature for speech recognition. Later, deep auto-encoder were used for speech recognition with MFCC as the input [24]. In one of the studies for using deep auto-encoder [25] they used the dysarthric speech. (Dysarthria is speech disorder, resulting in mumbled, slurred or slow speech). Dysarthic speech is generally difficult to understand for both humans and machines. In their studies they used MFCC representation with deep auto-encoders to get alternate representation for speech recognition. They were able to show an overall improvement of absolute 16% on Universal Access Dysarthic Speech Corpus.

Another strategy which was used to train deep auto-encoders was as follows with firstly the layer-wise training of the encoder part was done just like RBM pre-training. Later the parameters were fine-tuned using gradient descent as done in [26]. In this [26] they focused on de-noising and de-reverberation for noisy reverberant speech. They used the above stated strategy for training the auto-encoder. Their proposed approach shows that they were able to get significant improvement on recognition accuracy for various signal to noise ratio. Another work with similar training strategy was done in [27]. In [27], they used spectrograms as input to the system and were trying to get binary encoding of the speech for speech compression and rapid speech retrieval.

As in the case of RBM, where the scalability and parameter space became issues, the same problem were faced by auto-encoders. The modification to that was use of convolution operation instead of linear operation (Convolutional auto-encoders) [28].

Another advantage of using a convolution operation is it can deal with local temporal-spectral structures. Convolutional auto-encoders have been used for single channel audio source separation [29], music removal from speech [30], feature extraction, de-noising [31], etc. In [31] they were able to show that convolutional auto-encoder is able to learn pattern of different music genres. They used this auto-encoder for music removal from speech and also for feature extraction. In this work the authors have used convolutional auto-encoders for de-noising purpose. They have used spectrograms as the input to the system and tried generating the spectrograms back. They used ChiME data for evaluation of model. They studied the architectural properties of the convolutions like size of filters and number of filters to be used etc. They found that for spectrographic data large filter sizes and Tanh activations gained higher performance as compared to small filter size and linear rectifiers. Another thesis has done comparison of shallow convolution RBM and shallow convolution auto-encoder. Due to use of shallow networks they were not able to surpass the performance of MFCC. They used spectrograms as the input to all the models they studied.

Auto-encoder, RBM and all their variants gained popularity because they were able to give good initialization to deep networks which made it possible to train network. In 2011, the researchers found that as the layers increase, the pre-training does not help in performance improvement, but is very marginal [32]. And as large datasets and better hardware became available, researchers started with random initialization and were able to get the similar results as that of pre-training a network, if they train the network for long time. So the focus shifted towards directly learning representation instead of pre-training a network and then fine tuning it. There has been significant progress in training network from scratch as demonstrated in [4, 6, 7].

Initial work done in [33] used MFCC and Deep Neural Network (DNN) without any pre-training. They were able to get the similar results as that of with training. After DNN the researchers used Convolutional Neural Network (CNN) in the initial layer then used the fully connected to get better representation [34]. While all these

developments were done using MFCC representation the research for learning from raw wave form or spectrogram was also progressing. Some of the work using the raw waveform are [35, 36] and with spectrogram as input are [37, 38]. For the CNNs working on spectrogram one major advancement was limited weight sharing depending on the frequency [39]. In all the above architectures there were few convolutional layer followed by fully connected layer and recurrent layer to model the temporal behavior. Recent architecture developed for vision task show that using only convolutional layer we can get the comparable results for vision task [40]. As the fully connected layer is eliminated, the parameter space reduces significantly and also we are able to train very deep networks. These kind of architectures which do not a have fully connected layer try to exploit the representation learning power of CNN. After the removal of the fully connected layer from CNN major advancement in CNN architecture is addition of skip connection to help gradient propagation. A recent study shows that similar architecture can be used to model environmental sound and get comparative results using one dimensional CNNs [41]. In the same study it was the first time that a full 4sec waveform was fed to pure CNN and the CNN results were comparable to the MFCC based models.

Temporal variability can also be modeled using 1D CNNs. But the use of 1D CNNs for modelling temporal behavior is not explored much in the past for speech tasks. In natural language processing, time series analysis, medical signals 1D CNNs are used to model the temporal behavior. Majorly RNN and LSTMs are used for modelling temporal behavior in speech. We can reduce the computational complexity if we use 1D CNN instead of RNN.

In this chapter we discussed how the representation learning architecture evolved from 2006 to till date, in next chapter we discuss the building blocks and the experimental setup for the architecture used in our work.

CHAPTER 3

MODEL DESCRIPTION

3.1 Background

3.1.1 Convolution network

Convolutional networks are special networks which work on grid like structures. These grid like structure can be 1D structures like time-series data, 2D structures like gray scale images or 3D for color images. Convolutional network uses convolution operation instead of linear operation. Basic working of convolution operation is described in FC3.1.

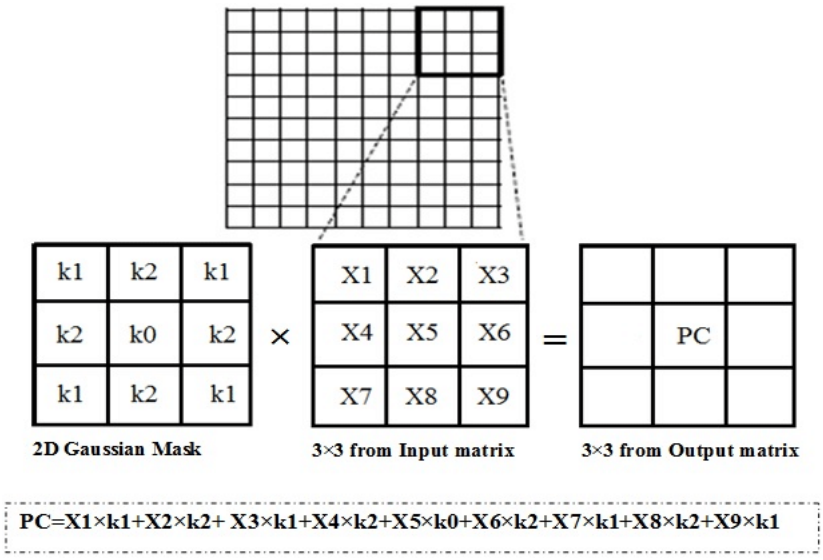

Figure FC3.1: Basic convolution operation

Characteristic of convolution network are as follows:

- Sparse interactions
- Parameter sharing
- Equivariant representation.

3.1.1.1 Sparse interactions

In feed forward network, a hidden layer neuron is connected to all the neurons of previous layer i.e. neurons are connected in densely manner. In convolution network the neuron is not connected to all the previous neuron but to some specific number of neurons. To be more specific, neurons in convolutional network act as a filter and the connectivity with previous layer depends on the size of filter. Due to filter like connectivity the neurons interact sparsely with input. This can be visualized from the FC3.2.

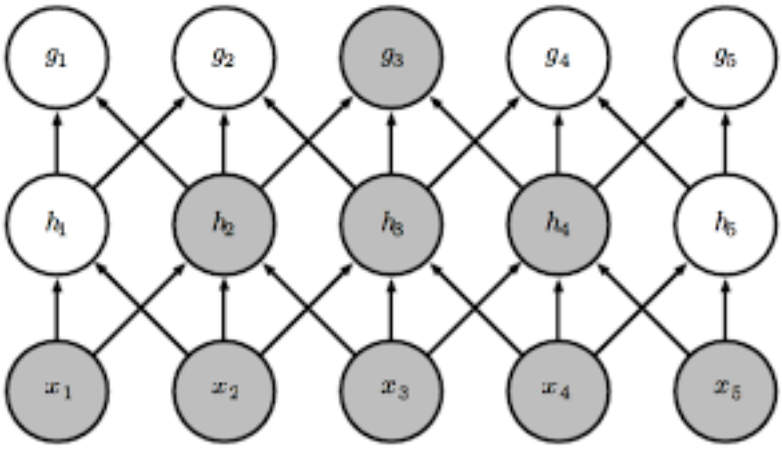

Figure FC3.2: Sparse Interaction

3.1.1.2 Parameter sharing

As discussed above the neuron acts as filter, and due to this same weights are applied at different locations in input which makes parameter sharing possible in convolution. This and sparse connectivity helps to reduce the parameter/weight space of the network and makes it more efficient as compared to typical feed forward network.

3.1.1.3 Equivariant representation

This is the outcome of above two properties of the convolution. As there are sparse connection and parameter sharing, weights learned for specific pattern can recognize that pattern at any input location. In the simplest form we can have translation variance, i.e. as the same filter will be applied to all the location so even if the pattern is shifted to some other location it will be detected by the same filter.

3.1.2 Non-linearity

Combination of multiple linear operation can be expressed as single linear operation which cannot handle non-linear pattern. To overcome this property of linear operation, we add non-linear activation function to extract complex feature after applying the filter/convolution on the input. Some examples of the non-linear activation functions are as follows.

- Sigmoid

  $f(x) = \frac{1}{1+e^{-x}}$

- tanh

  $f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

  tanh is basically a scaled version sigmoid.

- ReLU

$f(x) = max(0,x)$

As compared to sigmoid and tanh, ReLU is very efficient as it does not have exponent calculation and no division operation. There are several other variants of ReLU and one recent non-linear activation function is maxout introduced in [42].

3.1.3 Pooling

This operation is done after applying convolution and passing the output of convolution via non-linear activation. Pooling operation replaces the content of the specific window size with summary statistic. This phase is also called as detector phase or sub-sampling phase.

Basically after applying the convolution and non-linearity we have some features available for us, so we can afford sub-sampling of the input, so that the dimensionality can be reduce for further layers. Most commonly used pooling operation is max-pooling. Pooling operation helps makes the representation become more invariant.

3.1.4 Batch-Normalization

Normalization is a technique of scaling data to a specific range. This is done to give equal importance to all the variable/dimension of the data. This normalization is done to reduce the across dimension variability keeping the intra-class variability same. This is just shifting the mean to zero and scaling to unit variance. This helps the model to converge rapidly.

Such normalization is done before feeding the data to network but while training the network the across dimension variability may go up and affect the training procedure, so to overcome this batch-normalization [43] does the normalization of data for hidden layers; this helps us to train deep networks without the problems like vanishing/exploding gradients. It not only does the zero mean and unit variance to the data but also scales and shifts the data by some factor. The scaling and shifting factors are the learnable parameters of the network. This normalization and scaling is done on a mini-batch while

training. The algorithm can be summarized as in FC3.3.

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Figure FC3.3: BatchNorm with β and γ as learnable parameters

3.1.5 Residual Connection

Even with batch normalization, it becomes difficult to train very deep networks, so researchers added identity/skip connection inside the network so that this identity/skip connection helps the gradient flow to the initial layers. Another motivation to add this kind of connection was to learn the residual left after modelling some part. To put it formally, consider we are trying to model $\mathscr{H}(x)$ which is the original mapping and we are trying to approximate it. Suppose after stacked non-linear activation the network is model $\mathscr{F}(x)$ which can defined as follows

$\mathscr{F}(x) := \mathscr{H}(x) - x$

So the next layer has to model $\mathscr{F}(x) + x$ which can be realized by identity/skip connection. It can be visualized from the FC3.4.

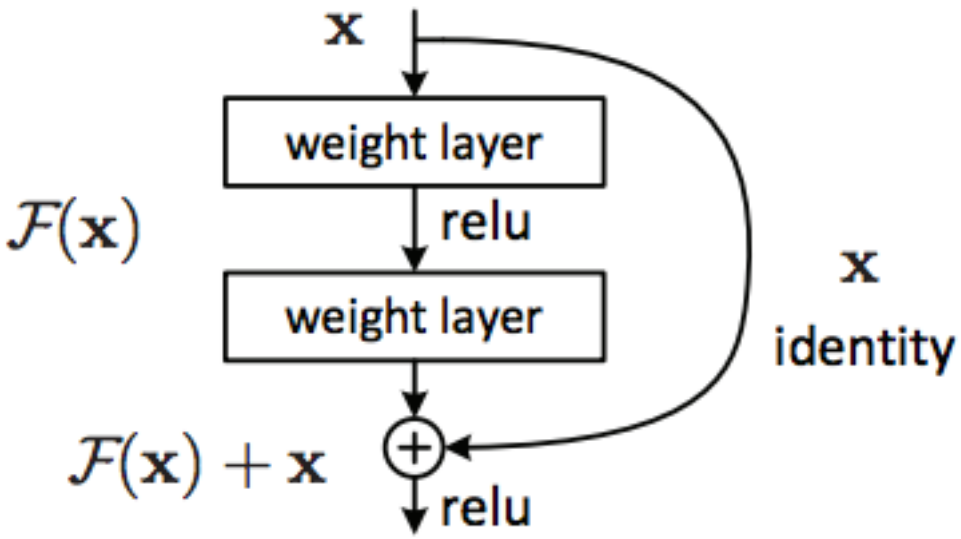

Figure FC3.4: Residual Connection

3.1.6 Connectionist Temporal Classification

This section introduces the Connectionist Temporal Classification (CTC) which is the output layer for neural networks. This output layer is used for temporal classification task i.e. for sequence labelling where the alignments between input sequence and the output sequence is not known. To overcome the alignment problem the CTC uses one extra label (say blank '_') with all the labels in the data. So to model this we add one extra unit in the last layer of the network before softmax activation. This extra label helps to overcome the problem of alignment of input and output sequence. The working of network with CTC loss can be explained as followed.

Consider we have segmented a waveform into 'n' units without worrying about the alignment with respective phoneme which are suppose 'm'. So now the network outputs 'n' labels corresponding to all the input segments, consider this as π which is one path. This π also contains '_' in it for some segments. The probability for one such π can be given as follows:

$$p(\pi|x) = \sum_{t=1}^{t=n} y_t$$

Now as per the CTC loss the duplicate labels will be deleted which are between the blanks and then all the blanks are deleted this collapsing function can be considered as $\mathscr{B}$, so we get a new output labels which is suppose 'l'. Now to calculate the probability of getting 'l' we consider the following equation.

$$p(l|x) = \sum_{\mathscr{B}^{-1}(l)} p(\pi|x)$$

In above formula sum is over $\mathscr{B}^{-1}(l)$ is used to get the all the path which outputs the same label and as these path are independent of each other we can sum these probabilities. By making use of blank and these mapping function the CTC overcomes the limitation of alignment as it considers all the paths which may output the same sequence.

3.2 Model architecture

Below given table TC3.1 describes the model used for all the experiments. Our architecture is adopted from recent work which shows comparative analysis of shallow, deep and very deep convolutional networks on environmental sounds [41]. We modified the architecture as per the given guidelines by the authors. The key idea behind these architecture is that they exploit the representational power of pure convolution network i.e. without using any kind of fully-connected layer. Inspired by their work we tried the similar architecture for speech signal. Some of the key component of the architecture are as follows.

3.2.1 Deep architecture

To exploit the representational power of convolution network and have comparison we have models with number of layers ranging from 3 to 34 layers. To train such deep networks we use small kernel size of 3. Only in the initial layer we use large kernel size of 160 which indicates 10ms if sampled at 16kHz this was suggested in paper. By this the initial layer is able to get 10ms and then builds complex features on and above it. Also the pooling is done only twice, this is possible because we are dealing with small input dimension and if we do pooling as suggested in the [41] we dont have anything till end. But as we go deeper the number of filters are increased to extract more and more complex features. We use rectified linear units for lower computation cost as used [44].

3.2.2 Fully convolutional network

Recent trends in vision tasks show that eliminating the fully connected layer, this reduces the parameter space significantly. It also helps us to know whether given convolutional layers have the capacity to learn discriminative features. Using the same idea from vision task we eliminated the fully-connected layer and used global average pooling as used in [40] for final classification. By removing the fully-connected layer the system is forced to learn good representation which potentially generalizes better.

3.2.3 Experimental Setup

We use TIMIT data set for all the experiments which has 462 speakers for training and 168 speakers for testing. The number of utterances in training is 3696 and in complete test set is 1344 utterances. TIMIT has 61 phoneme classes which are mapped to 39 phoneme classes as suggested in [45]. The merged phonemes are as follows.

$aa, ao \rightarrow aa$

$ah, ax, ax-h \rightarrow ah$

$er, axr \rightarrow er$

$hh, hv \rightarrow hh$

$ih, ix \rightarrow ih$

$l, el \rightarrow l$

$m, em \rightarrow m$

$n, en, nx \rightarrow n$

$ng, eng \rightarrow ng$

$sh, zh \rightarrow sh$

$uw, ux \rightarrow uw$

$pcl, tcl, kcl, bcl, dcl, gcl, h\#, pau, epi \rightarrow sil$

$q \rightarrow -[deleted]$

Table TC3.1: Architecture of CNNs

M3	M5	M11	M18	M34 resnet
Input n x1 time domain waveform				
[160/4,256]	[160/4,128]	[160/4,64]	[160/4,64]	[160/4,48]
maxpool : 4x1				
[3,256]	[3,128]	[3,64]x2	[3,64]x4	$\begin{bmatrix} 3 & 48 \\ 3 & 48 \end{bmatrix}$ x3
maxpool: 4x1				
	[3,256]	[3,128]x2	[3,128]x4	$\begin{bmatrix} 3 & 96 \\ 3 & 96 \end{bmatrix}$ x4
	[3,512]	[3,256]x3	[3,256]x4	$\begin{bmatrix} 3 & 192 \\ 3 & 192 \end{bmatrix}$ x6
		[3,512]x2	[3,512]x4	$\begin{bmatrix} 3 & 384 \\ 3 & 384 \end{bmatrix}$ x3
Global average pooling				
Softmax				

Architectures of adopted fully convolutional network for time-domain waveform inputs. M3 denotes 3 weight layers. [160/4, 256] denotes a convolutional layer with receptive field 160 and 256 filters, with stride 4. Stride is omitted for stride 1 (e.g., [3,

256] has stride 1). [...] k denotes k stacked layers. Double layers in a bracket denotes a residual block and only occur in M34-res. All convolutional layers are followed by batch normalization layers, which are omitted to avoid clutter.

3.2.4 Architectural Modification

One major challenge adopting the architecture from [41] was input dimension, as they had very large input size of 32000 dimension as compared to 800 dimension. So we modified the architecture as follows

- Changed the kernel length from 80 to 160 depending on sampling frequency.
- Removed the pooling layers to make data available for deeper layers.
- Added batch norm layer as the first layer i.e. immediately after the input so that network learns an appropriate scale of the data.
- Other than above mentioned changes we tried changing the stride size, kernel size, pooling window and position of pooling layer from the adopted architecture.

In this chapter we discussed the building blocks needed to understand this work. We also described the architecture of CNNs used for experimentation with all the modifications. In the next chapter we discuss the results and key findings from all the experimentation setup done in this chapter.

CHAPTER 4

RESULTS

This chapter discusses the results of various experiments done in this thesis. The results are majorly divided into two section, one is frame-level and other is continuous classification.

4.1 Frame-level model

4.1.1 Different length inputs

Experiments were carried out for various length input like 10ms, 25ms, 50ms & 100ms and the input dimension at 8kHz was 160, 400, 800 and 1600 respectively. As the same phoneme unit was divided for training the number of data-points in each experiment was different. For input dimension 1600 i.e. 100ms there were very few train and test examples because of which the models M5, M11 and M18 showed

Table TC4.1: Results with varying input length

Model/Dim	160	400	800	1600
M3	57.15	60.13	61.38	66.73
M5	57.35	61.96	64.46	Unstable
M11	57.50	63.11	66.27	Unstable
M18	58.10	63.59	66.92	Unstable

From the table TC4.1 we can clearly conclude that the larger the input size, more is the gain for the model. But the problem here is that the average length of phoneme is around 70ms so large input dimension will be not be available for training.

4.1.2 Normalization

In table TC4.2 we provide the results with various normalization techniques to reduce the variability that CNNs needs to deal with if raw-waveform is fed.

Table TC4.2: Normalization Results

Model/Dim	Gain	Pitch	Time	Batch
M3	61.38	54.39	62.16	68.28
M5	64.46	58.02	65.72	71.31
M11	66.27	61.04	69.06	71.13
M18	66.92	62.22	69.94	70.92
M34	67.10	58.43	67.64	70.84

In table TC4.2 we see that batch normalization gets the best results when compared to all the other normalizations done. This is one of the major result from our work that instead of feeding zero mean and unit variance data to network, we can feed raw data and keep first layer as the batch norm layer which helps to scale and shift the data as needed instead of zero mean and unit variance.

4.2 CTC based models

The results with CTC loss and decoding are given in the table TC4.3. As we saw from the previous results that 800 dimension input was performing well on the architecture we used the 800 dimension input for the CTC model. We also used the batch normalization layer as the first-layer so that it performs well as observed from the normalization result table.

Table TC4.3: CTC results

Model	Phoneme Error Rate
M3	53.60
M5	47.78
M7	43.81
M9	39.94

For the models with CTC loss we can see that as the layers are being added the phoneme error rate goes down. The best model was M9 which had 9 convolutional layers was able to get accuracy of 60.06%.

In this chapter we discussed the results of all the experiments done in this thesis. In next chapter we will discuss the future work that can be done to enhance the quality of work.

CHAPTER 5

FUTURE WORK

- Training deep CNNs with CTC loss

  In our experiments we have trained network with nine layers with CTC loss. And as observed from the results for CTC loss we see that the performance of the model increases so we can try very deep models and observe the results for the same.

- Adding language model and word-level decoding

  Currently the model is able to do continuous decoding at phoneme level. We can add language model and do word level decoding, which helps to generate the transcript of given input.

- Studying other normalization technique and results.

  Various other normalizations like channel normalization or speaker normalization can be done and how it affects the performance of the model can be studied. Other methods of pitch normalization can be studied and check its effect. Effects of combining various normalization techniques with each other can be studied.

- Visualizing and understanding what system is learning. Interpreting the weights and visualizing them is still an open problem for CNNs in speech research. Such studies can be performed so that what the CNNs have learned can be understood.

CHAPTER 6

CONCLUSIONS

The main aim of this work was to exploit the representation power of CNN for speech waveform. Various 1D CNNs architectures were used for this experiment. We believe that applying CNNs would help to reduce the complexity of the system but would get the state of the art results by only applying CNNs.

Initial aim was have raw waveform as the input, but as discussed raw waveform models does not converge, so need to have some basic normalization to reduce the variability. Our experiments show that we can gain significantly by applying normalization to waveform over raw waveform. Maximum improvement was via duration normalization and using batch normalization layer as the first layer.

One of the findings of the experiments is that the input dimension affects the saturation of the networks. Lesser the input dimension earlier the networks representation power saturates. Although this can be overcome by using fully-connected layer to add the discriminative power to the network. But if we use only convolutional layers then input dimension is a crucial factor for getting results.

Overall in this study we explored the representational power of CNNs for speech waveform. We performed various experiments related to normalization and input dimension and conclude that deep CNNs with some normalization can be used as an alternate to the traditional MFCCs in an end-to-end setup.

Bibliography

[1] Sadaoki Furui. Fifty years of progress in speech and speaker recognition. *The Journal of the Acoustical Society of America*, 116(4):2497–2498, 2004.

[2] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony Han, Lappi Vaino Johannes, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, and Zhenyao Zhu. Deep speech 2: End-to-end speech recognition in english and mandarin. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 173–182. JMLR.org, 2016.

[3] Yann Lecun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[5] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[7] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *ICLR 2016 Workshop*, 2016.

[8] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 369–376, New York, NY, USA, 2006. ACM.

[9] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren. Darpa timit acoustic phonetic continuous speech corpus cdrom, 1993.

[10] Geoffrey E. Hinton and Simon Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006.

[11] A. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *Trans. Audio, Speech and Lang. Proc.*, 20(1):14–22, January 2012.

[12] Abdel rahman Mohamed, George Dahl, and Geoffrey Hinton. Deep belief networks for phone recognition. In *in Proceedings of the NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[13] George Dahl, Marcaurelio Ranzato, Abdel rahman Mohamed, and Geoffrey E Hinton. Phone recognition with the mean-covariance restricted boltzmann machine. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 469–477. Curran Associates, Inc., 2010.

[14] Navdeep Jaitly and Geoffrey E. Hinton. Learning a better representation of speech soundwaves using restricted boltzmann machines. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5887. IEEE, 2011.

[15] O. Ghahabi and J. Hernando. Deep belief networks for i-vector based speaker recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1700–1704, May 2014.

[16] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 609–616, New York, NY, USA, 2009. ACM.

[17] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1096–1104. Curran Associates, Inc., 2009.

[18] Pierre Baldi. Autoencoders, unsupervised learning and deep architectures. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27*, UTLW'11, pages 37–50. JMLR.org, 2011.

[19] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[20] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. Speech enhancement based on deep denoising autoencoder. In *INTERSPEECH*, 2013.

[21] S. S. Wang, H. T. Hwang, Y. H. Lai, Y. Tsao, X. Lu, H. M. Wang, and B. Su. Improving denoising auto-encoder based speech enhancement with the speech parameter generation algorithm. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 365–369, Dec 2015.

[22] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, December 2010.

[23] Jonas Gehring, Yajie Miao, Florian Metze, and Alexander H. Waibel. Extracting deep bottleneck features using stacked auto-encoders. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3377–3381, 2013.

[24] Takaaki Ishii, Hiroki Komiyama, Takahiro Shinozaki, Yasuo Horiuchi, and Shingo Kuroiwa. Reverberant speech recognition based on denoising autoencoder. In *INTERSPEECH*, 2013.

[25] Bhavik Vachhani, Chitralekha Bhat, Biswajit Das, and Sunil Kumar Kopparapu. Deep autoencoder based speech features for improved dysarthric speech recognition. In *INTERSPEECH*, 2017.

[26] Xue Feng, Yaodong Zhang, and James R. Glass. Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1759–1763, 2014.

[27] Li Deng, Michael L. Seltzer, Dong Yu, Alex Acero, Abdel rahman Mohamed, and Geoffrey E. Hinton. Binary coding of speech spectrograms using a deep auto-encoder. In *INTERSPEECH*, 2010.

[28] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I*, ICANN'11, pages 52–59, Berlin, Heidelberg, 2011. Springer-Verlag.

[29] Emad M. Grais and Mark D. Plumbley. Single channel audio source separation using convolutional denoising autoencoders. *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1265–1269, 2017.

[30] M. Zhao, D. Wang, Z. Zhang, and X. Zhang. Music removal by convolutional denoising autoencoder in speech recognition. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 338–341, Dec 2015.

[31] Victor Zhong Mike Kayser. Denoising convolutional autoencoders for noisy speech recognition. , Tech Report, Stanford University, 2015.

[32] Dong Yu and Li Deng. *Automatic Speech Recognition: A Deep Learning Approach*. Springer Publishing Company, Incorporated, 2014.

[33] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov 2012.

[34] Ossama Abdel-Hamid, Abdel-Rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 22(10):1533–1545, October 2014.

[35] Dimitri Palaz, Ronan Collobert, and Mathew Magimai-Doss. Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks. In *INTERSPEECH*, 2013.

[36] Tara N. Sainath, Ron J. Weiss, Andrew W. Senior, Kevin W. Wilson, and Oriol Vinyals. Learning the speech front-end with raw waveform cldnns. In *INTERSPEECH*, 2015.

[37] Ying Zhang, Mohammad Pezeshki, Philmon Brakel, Saizheng Zhang, Csar Laurent, Yoshua Bengio, and Aaron Courville. Towards end-to-end speech recognition with deep convolutional neural networks. In *INTERSPEECH 2016*, pages 410–414, 2016.

[38] Cornelius Glackin, Julie A. Wall, Gérard Chollet, Nazim Dugan, and Nigel Cannings. Convolutional neural networks for phoneme recognition. In *International Conference on Pattern Recognition Applications and Methods*, pages 190–195. SciTePress, 2018.

[39] Tara N. Sainath, Brian Kingsbury, Abdel-rahman Mohamed, George E. Dahl, George Saon, Hagen Soltau, Tomás Beran, Aleksandr Y. Aravkin, and Bhuvana Ramabhadran. Improvements to deep convolutional neural networks for LVCSR. *CoRR*, abs/1309.1501, 2013.

[40] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[41] Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. Very deep convolutional neural networks for raw waveforms. *CoRR*, abs/1610.00087, 2016.

[42] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1319–III–1327. JMLR.org, 2013.

[43] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015.

[44] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress.

[45] Kai-Fu Lee and Hsiao-Wuen Hon. Speaker-independent phone recognition using hidden markov models. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 37:1641–1648, 1989.