Table of Contents

[Background 2](#_Toc490870777)

[Introduction 2](#_Toc490870778)

[An Overview of Automatic Speech Recognition 2](#_Toc490870779)

[Previous Work 3](#_Toc490870780)

[Methodology 4](#_Toc490870781)

[De-noising and Enhancement of Speech 4](#_Toc490870782)

[Feature Extraction 5](#_Toc490870783)

[Deep Neural Network 9](#_Toc490870784)

[Activation Function 10](#_Toc490870785)

[Multi-Layered Feed-forward Network 12](#_Toc490870786)

[Backpropagation 13](#_Toc490870787)

[Optimization Algorithm 13](#_Toc490870788)

[The Cost Function 14](#_Toc490870789)

[Specifications for the experiment 14](#_Toc490870790)

[Hidden Markov Model with Gaussian Emission 15](#_Toc490870791)

[Estimating the parameters 16](#_Toc490870792)

[Decoding 16](#_Toc490870793)

[Reference 17](#_Toc490870794)

# 

# Background

## Introduction

------------------------------------------------------------------------------------------------------------------------------------------

## An Overview of Automatic Speech Recognition

Automatic Speech Recognition (ASR) is the process of identifying and responding to the distinct sounds produced by human speech. It enables a program or a machine to convert the spoken words or sentences in machine readable format and identify them. Automatic speech recognition systems can be classified in the following types:

1. **Speaker Dependent:** The system can recognize the speech of a single speaker. These systems are easier to develop, cheaper and more accurate but not as flexible and adaptive to practical applications for multiple speakers.
2. **Speaker Independent:** The speaker independent system is developed to work with any speaker. These systems are more complex, expensive and have less accuracy but more flexible and adaptable for practical applications that utilize speech recognition for multiple speakers.

Speech recognition can also be classified in the following types:

1. **Isolated Word Recognition:** The words recognized by such systems are separated by pauses or includes utterances of one single word at a time. These are not surprisingly easier to construct because the end points of the speech signals are easily detectable and the pronunciation of a word does is not affected by other words. They can also be quite robust since all possible patterns for the inputs are known. Isolated word recognition systems can be designed and built for certain application oriented words such as, digits recognition for phone dialing, navigation related words e.g. left, right, forward, backward etc.
2. **Continuous Speech Recognition:** A continuous speech recognition system operates on words that are connected without pauses. It recognizes the natural flow of speech. The increased complexity of such systems arises because of a number of factors. First, it requires detection of start and end points of each word. Another problem is that, since the phonemes are connected together, utterance of each word is affected by its surrounding words. This is known as “co-articulation”. It is also affected by the speed and rate of speech.

Some speech recognition systems may only need to recognize a few words, for example the digits while others need a large set of words depending on the application. Automatic speech recognition systems can be further classified in the following way base on the size of the vocabulary used and recognized by the system:

1. **Small Vocabulary:** Usually works with less than a hundred words. It is possible to get quite an accurate result for speaker independent systems with small vocabulary. Applications such as, voice interface for phone dialing, navigation of robots, operating smartphones and so on.
2. **Medium Vocabulary:** Systems that uses medium sized vocabulary usually work with a set of 1000-3000 words.
3. **Large vocabulary:** These systems use thousands or tens of thousands of words (e.g. 20000 words). It is difficult with a large vocabulary dataset to attain a satisfactory level of accuracy. They usually need to be speaker dependent to get a higher accuracy.

## Previous Work

------------------------------------------------------------------------------------------------------------------------------------------

# Methodology

In this chapter, the dataset used and the theories behind HMM with Gaussian Emissions and DNN which are used as classifiers are explained. ASR for isolated words has three main steps:

1. Preprocessing
2. Feature Extraction
3. Classification

In each step, the theories behind it and the experimental configuration are described.

## De-noising and Enhancement of Speech

The first step in ASR process is preprocessing. It is necessary to de-noise and enhance the raw speech data before it goes through any further signal processing. The raw speech data can be corrupted by three kinds of noise:

1. Recording noise
2. Electrical noise
3. Environmental noise

The first two types of noises can be easily compensated by training the system with data that has similar noise. However, the third one can severely degrade the performance of the recognition because of its varying nature. The challenge of reducing noise is reducing external noise without affecting the low-intensity components of the speech [17].

All word samples were put through five stages of enhancement. First, stereo channels were merged into a single mono channel. Then, static background noise was attenuated using a noise-reduction algorithm based on Fourier analysis. Unique noise profiles were used for each word samples for best results.  Then, the sound signals were normalized to have maximum amplitude of -1.0 dB and 0 mean amplitude displacement for uniformity. Any silence at the beginning or end was truncated. Finally, the audio samples were cloned into two separate datasets, one of which was then synthetically augmented by including pitch altered voice samples from existing data. The effect of these five stages on a particular audio sample representing the Bangla word for “first” can be seen in fig. 1. The end result is a concise audio sample that is ready to be used to train and test different acoustic models.

|  |
| --- |
|  |

## Feature Extraction

Feature extraction is the first step in Automatic Speech Recognition which encodes the speech data as a set of quantifiable feature vectors that are fed into the acoustic models. Speeches are certain sounds that are shaped by articulation of the vocal cord, nose, tongue, teeth and other organs. Speech signal is basically a one dimensional waveform which has some discrete and measureable qualities to it, such as, energy level, certain frequencies and so on. It is important to select feature vectors in a way that minimizes redundancy, gets rid of unwanted or unimportant features and focuses on features that distinctly represent different sample classes.

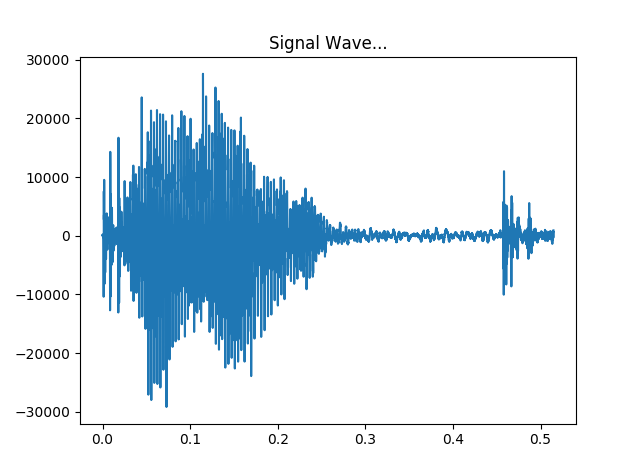


Fig 1: Waveform of a spoken word

Mel-Frequency Cepstral Coefficients (MFCC) is the most popular feature vectors for speech and voice recognition. Paul Mermelstein and S. B. Davis [1] [2] are accredited for this idea in the 1980’s and Mel Frequency Cepstrum has been the state-of-the-art till date. The popularity of MFCC vectors is attributed to the fact that it closely mimics the way human ears perceive sound and respond accordingly. MFCCs are commonly derived in the following steps [3, 4, and 5]:

Speech is a non-stationary time varying signal, i.e. it is always changing. Therefore, the signal is divided into small segments, e.g. 20-30ms frames where the signal can be considered to be statistically stationary. If the frame is too long, the signal may change too much which will cause the feature vectors to be a lot less useful for accurate predictions. Frame steps are 10ms which causes some overlapping of frames. The frames are 10ms long. As a result there exists some overlapping between two consecutive frames.

After dividing the signals into small frames, the second step is calculating the power spectrum of each frame. This was inspired by the mechanism of cochlea, an organ inside human ear. It vibrates at different spots depending on the frequency which in turn fires different neurons. The periodgram spectral estimates similarly detect the frequencies present in the input signal. This is done by taking the Discrete Fourier Transform (DFT) of each step. The DFT is represented by the following equation:

XN [n] = XN [n+N]

Here, XN[N] is a signal with a period of N. If the DFT is XN[n], k is the length of the DFT,

XN[k] =

XN[n] =

The periodgram estimate of power spectrum is given by:

PN[k] = [k] |2

257 out of 512 point FFT are kept.

|  |
| --- |
|  |

Figure: Power Spectral Density of the word “ak”

The initial periodgram estimated contain a lot of information that are unnecessary for Automatic Speech Recognition. The cochlea cannot effectively differentiate between two closely spaced frequencies. This is why the Mel-spaced Filter bank is used. The first filter grows from narrow to wider as it provides an estimate of the amount of energy present near zero hertz to higher frequencies. 20-40 (here 26 is used) triangular spaced filters are applied to the periodgram power estimates. This results in 26 vectors of length 257. Each filter bank is multiplied with the power spectrum and then the coefficients are added up to calculate the filter bank energies. The non-linear frequency scale used her is called mel scale and it helps to mimic how human ear respond to frequencies. It is given by the following equation:

M(f) = 1125 ln (1 + f/700)

In the next step, the logarithm of the filter bank energies is taken. This is also influence by how humans perceive sound. It is necessary for the large variation of energy to sound similar. So the perceive loudness is increased so that the variations are insignificant since the sound was already loud enough compared to the energy variations to make it ignorable. Therefore, the logarithm of each of the 26 energies from the previous step is taken.

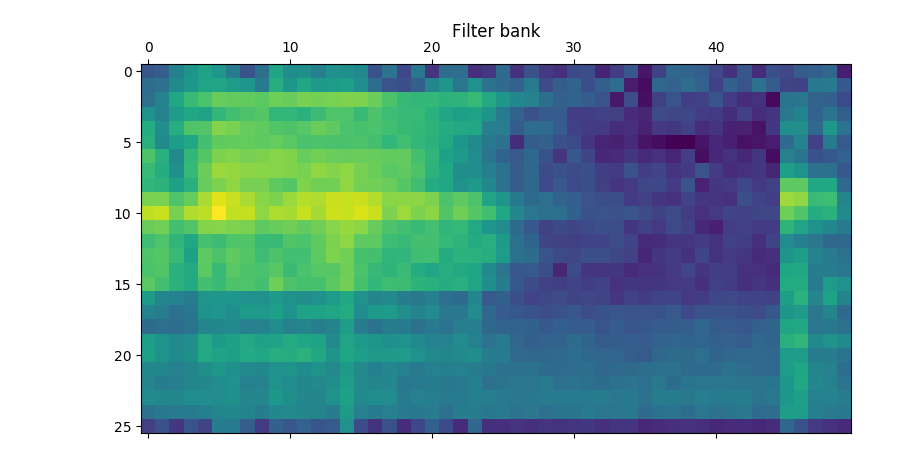


Figure. Log Filter bank

The next step is calculating the Discrete Cosine Transform (DCT) of the 26 log filter bank energies from the previous step. The filter bank energies are correlated to each other since they are overlapping. This step fixes that so that these can be used for diagonal covariance matrices, e.g. the ones used for Hidden Markov Model based classifiers. This step results in 26 cepstral coefficients only the lower 13 of which are kept.

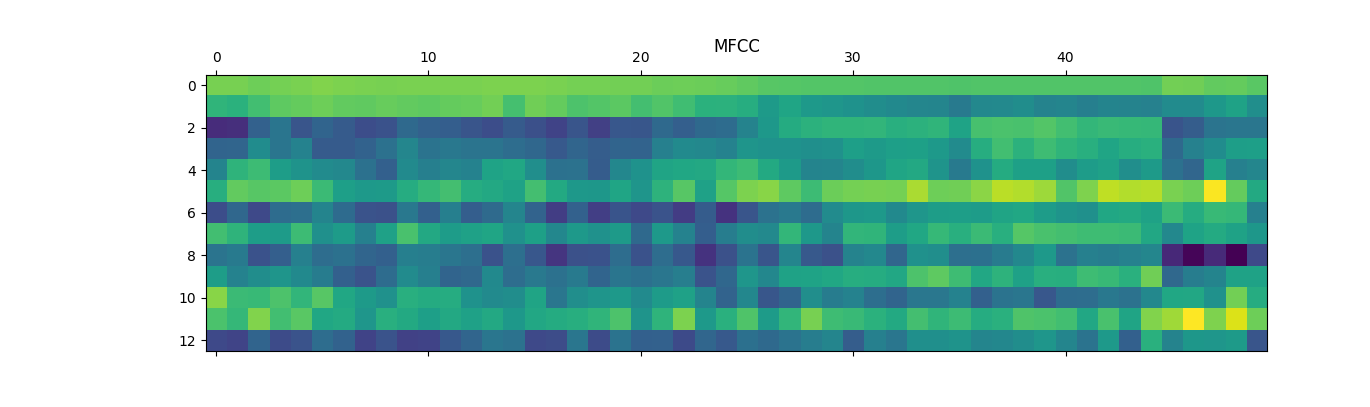


Figure. Extracted MFCC features

## 

## Deep Neural Network

Deep Neural Network has proven to be successful in speech recognition and is currently a widely researched area under this field [8, 9, and 10].

Artificial neural networks are simplified representation and simulation of the neuronal structure present in brains. Deep neural networks are artificial neural networks where multiple layers of neuron are used. To explain the analogy of how they work, neurons in brain can receive signals and there are communication links from one neuron others. A neuron can respond to input signals. If the strength of the signal is above a certain level, the particular neuron is activated or fired. The output of any neuron can be the input to other neurons. There is a feedback involved in the process depending on the result produced by activating a neuron. If the result is the desired output, the connection that caused that result is strengthened. The system learns through observations and the feedback mechanism.

The stated concept of neuronal structure is mimicked in Artificial Neural Network. The training of an Artificial Neural Network system can be of two types. First, supervised learning where a set of labeled input data is used for trainings. The labels are the desired outputs. On the other hand, in unsupervised learning, the network learns the structure of the given input and learns through feedback for the outcome it produces.

### Activation Function

An artificial neuron is called a perceptron. The idea if perceptron was brought forth by [Frank Rosenblatt](http://en.wikipedia.org/wiki/Frank_Rosenblatt) in the 1950s and 1970s which was inspired by the earlier work of [Warren McCulloch](http://en.wikipedia.org/wiki/Warren_McCulloch) and Walter Pitts. However, different models for neurons are now being used one of which is the Sigmoid Neuron. [6]

As shown in the figure below, a perceptron receives inputs (X1, X2………Xn) and generates a binary output (Y). The output is a function of the input and weights (W1, W2…..Wn). Weights are variables that change during the learning process. There is a certain threshold value. If the weighted sum of the inputs is greater than the threshold, then the output of the perceptron is 1, or the perceptron will fire. Else, the output is 0. The function f(X) is the activation function for a perceptron.

X1

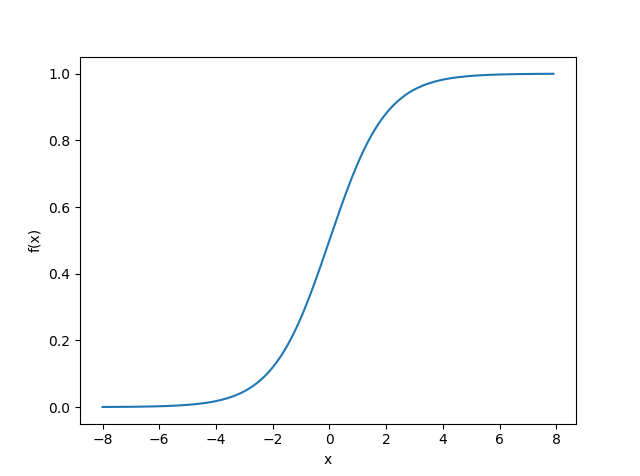
f(X)

X2

Xn

On the other hand, for a sigmoid neuron, the sigmoid function works as the activation function. The sigmoid function is a widely used function for feed-forward network with back propagation because of its non-linearity and simplicity of computation [11]. The function is given by:

f(X) =

Here h(x) is the input. The function generates the following curve:  
 

In practical application of using the sigmoid function as the activation function, Wi is real valued weight, Xi is the input then the weighted input of a nod is given by:

g(X) = X1W1 + X2W2 +…..+ XiWi +…..+ XnWn + b

The weight variable is changed depending on the how much the relationship between the inputs to the output need to be strengthened. As the figure shows, weights control the slope of the sigmoid function and the bias controls when the node activates. If x is greater than a certain value, the node can be activated. Biases ensure that condition.

The following figure shows the effect of different weight to the sigmoid function:

|  |  |
| --- | --- |
| (a) |  |

Fig. (a) Effect of different weights on sigmoid function (b) Effect of biases on the sigmoid function

### Multi-Layered Feed-forward Network

Feed-forward network is the type of Artificial Neural Network where the connections between the nodes do not form a cycle [7]. A simple three layered feed-forward neural network structure is shown below:

|  |
| --- |
| b  b  H(x)  H3(2)  H2(2)  H1(2)  X3  X2  X1 |

Figure. A three layered neural network

The network can be defined as follows:

H1(2) = f( W11(1)X1 + W12(1)X2 + W12(1)X2 + b1(1))

H2(2) = f( W21(1)X1 + W22(1)X2 + W22(1)X2 + b2(1))

H3(2) = f( W31(1)X1 + W32(1)X2 + W32(1)X2 + b3(1))

H(x) = f( W11(2) H1(2) + W12(2) H2(2)  + W12(2) H2(2)  + b1(2))

Here Wij(L)represent the weight associated with the connection of node number i to j where j is in layer L and I is in layer L+1.The notation bi(L) stands for the bias weight in node number i in layer L+1. Finally, H(x) is the output of the last node in layer 3. The third layer node takes the output of the nodes in the second layer and thus there exists hierarchical network of nodes.

### Backpropagation

Backpropagation is the process of minimizing the differences between actual output and the desired output or the error based on the training samples with labels. This is used by optimization algorithms for adjusting the weight for each connection between neurons or nodes in different layers based on the accumulated error of a batch in the training data. The goal of backpropagation is to refine the mapping of inputs to outputs. The error is computed using a cost function which is propagated back through the network. Different optimization algorithms are used for the process.

### Optimization Algorithm

As stated earlier, the weights associated with the connections between the nodes in different hierarchical layers decide the strength of the relationship between the inputs and outputs. The target is to minimize the error. An error occurs when an input does not produce the desired output. In supervised learning, the system is trained with data were for each input the correct output is known. In the optimization process, the weights are varied to minimize the error based on the given samples during the training. It is also necessary that the optimization do not over-fit to the training data and can generalize well for the unseen test data. The objective of an optimizer is to get to the minimum point of the error curve for different weights. There are different optimization techniques such as Stochastic Gradient Descent, Limited memory BFGS, Conjugate Gradient and so on[12]. For our experiment, Adam, a stochastic optimization method was used which combines the advantages of two popular methods AdaGrad (Duchi et al., 2011) and RMSProp (Tieleman & Hinton, 2012). This technique was first introduced in 2014[13]. It takes the following parameters:

* **Learning rate**: A floating point value. The learning rate.
* **beta1**: A float value or a constant float tensor. The exponential decay rate for the 1st moment estimates.
* **beta2**: A float value or a constant float tensor. The exponential decay rate for the 2nd moment estimates.
* **epsilon**: A small constant for numerical stability. Float >= 0. Fuzz factor.

### The Cost Function

As mentioned before, the system minimizes the error iteratively by varying the weight with an optimization technique. A way of generalizing this process by not overfitting it to the training set is using cost function. It is a measure of how good a neural network does based on the training samples. For the experiment, softmax cross entropy with logits were used. It measures the probability error in discrete classification where each sample can belong to exactly one class.

### Specifications for the experiment

The Deep neural network used for the experiment has total four layers including an input layer, an output layer and three hidden layers. Each hidden layer had 1500 nodes. For the cost function, softmax cross entropy with logits was used and for optimizer Adam optimizer [13] was used.

## Hidden Markov Model with Gaussian Emission

Hidden Markov Models have been successfully used for time varying sequences such as audio signal processing. The underlying idea behind the Hidden Markov Model is that, it models sequences with discrete states. The way this maps to the problem of speech recognition is, during the feature extraction process, speech signals are transformed into features of discrete time slices or frames. Therefore there are a finite number of frames in a particular word. When a particular sequence of features is given, the model can yield the probability of that sequence being a certain word. Here, the phonemes i.e. the distinct units of sound that can be produced are discrete states and the sequences of MFCCs which represent the uttered word are observations. The probability of observing MFCC sequences given the state is performed using Gaussian Emissions.

The details [14] of how it works are stated below:

Hidden Markov Model is a probabilistic model which can produce a sequence of observation X by a sequence of hidden states Z. It is generated by a probabilistic function associated with each state.

An HMM is usually represented by λ where λ = (A, B, Π). It can be defined by the following parameters:

O = {o1, o2…………om}. This is an output observation sequence. For speech recognition, this represents the MFCC feature vectors.

Ω = {1, 2,………,N}. This is a set of states. For speech recognition, it is the phoneme labels.

A = {aij}. This is the transition probability matrix. It represents the probability associated with transition from state i to state j.

B = {bi(k)}. It is the output probability, i.e. the probability of emitting a certain observation ok in the state i.

Π = Start probability vector.

There are three basic problems for HMM:

1. Estimating the optimal sequence of states given the parameters and observed data.
2. Calculating the likelihood or probability of a data given the parameters and observed data P (O| λ).
3. Adjusting the parameters given the observed data so that P(O| λ) is maximized.

For isolated word recognition, for each word in the vocabulary, a separate N-state HMM is designed. For each word, the model is trained with feature sequences (MFCCs) of multiple utterances of a single word by different people. This is represented by problem 3 which is estimating the optimal model parameters. The solution is manifested by problem 1. Here, the frames of each word in the vocabulary (as described under feature extraction) are a state and the properties of the model are evaluated based on the corresponding MFCCs that led to that observation. Word recognition is performed by solution to problem 2 where the likelihood of an observed sequence of MFCC is calculated given the model parameters.

### Estimating the parameters

The solution to problem 3 is the most difficult one since there is no analytical method to maximize the probability in the training data. The idea is to estimate the model parameters so that P (O| λ) is maximized for the training observations. The optimal Gaussian mixture parameters for a given set of observations can be chosen such that the probability reaches maxima by using the Baum-Welch algorithm or Expectation Maximization (EM) algorithm [15]. It is a gradient based optimization method which is likely to converge at the local maxima.

### Decoding

Estimating the state sequence S given an observation sequence X and the model λ is done using the Viterbi algorithm [16]. It is a formal technique for finding the best state sequence based on dynamic programming method [14].

# Discussion:

Despite using HMM-GMM, traditional and proved method HMM-GMM and the state-of-the-art method, DNN for Isolated Word Recognition, the models achieved a low accuracy score for speaker independent system. The most probable reason behind that is, there was no available corpus for Isolated word Recognition in Bengali language to work with. The size of the corpus and the sparseness of the training data are what likely affected the performance of the classifiers. This assumption is supported by positive effect of increasing dataset on the accuracy score.

Despite the limitations of working with a small dataset, the experiment shows that improvements can be made by augmenting the data. This is due to the fact that, when the existing training data are augmented by changing frequency and/or length, it essentially mimics different utterances by more people and provides a representation of the variations in speech, thereby improving the performance.

The experiment further shows that the accuracy of HMM-GMM classifier drastically increases up to 96.67% in speaker dependent system. However, the DNN classifier shows no improvement. This again is attributed to the dependency of DNN on the amount of training data.

# Reference

1. P. Mermelstein (1976), "Distance measures for speech recognition, psychological and instrumental," in *Pattern Recognition and Artificial Intelligence*, C. H. Chen, Ed., pp. 374–388. Academic, New York.
2. S.B. Davis, and P. Mermelstein (1980), "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), pp. 357–366.
3. X. Huang, A. Acero, and H. Hon. Spoken Language Processing: A guide to theory, algorithm, and system development. Prentice Hall, 2001.
4. Min Xu; et al. (2004). "HMM-based audio keyword generation". In Kiyoharu Aizawa; Yuichi Nakamura; Shin'ichi Satoh. Advances in Multimedia Information Processing – PCM 2004: 5th Pacific Rim Conference on Multimedia (PDF). Springer. ISBN 3-540-23985-5.
5. Sahidullah, Md.; Saha, Goutam (May 2012). "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition". Speech Communication. **54** (4): 543–565. [*doi*](https://en.wikipedia.org/wiki/Digital_object_identifier):10.1016/j.specom.2011.11.004.
6. Michael A. Nielsen, “*Neural Networks and Deep Learning”*, Determination Press, 2015.
7. Zell, Andreas (1994). Simulation Neuronaler Netze [Simulation of Neural Networks] (in German) (1st ed.). Addison-Wesley. p. 73. [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*3-89319-554-8*](https://en.wikipedia.org/wiki/Special:BookSources/3-89319-554-8).
8. Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdelrahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” Signal Processing Magazine, IEEE, vol. 29, no. 6, pp. 82–97, 2012
9. George E Dahl, Dong Yu, Li Deng, and Alex Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” Audio, Speech, and Language Processing, IEEE Transactions on, vol. 20, no. 1, pp. 30–42, 2012.
10. Frank Seide, Gang Li, and Dong Yu, “Conversational speech transcription using context-dependent deep neural networks.” in INTERSPEECH, 2011, pp. 437–440.
11. Han, Jun; Morag, Claudio (1995). "The influence of the sigmoid function parameters on the speed of backpropagation learning". In Mira, José; Sandoval, Francisco. [*From Natural to Artificial Neural Computation*](http://dx.doi.org/10.1007/3-540-59497-3_175).
12. Ngiam, Jiquan, et al. "On optimization methods for deep learning." *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011.
13. Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).
14. Lawrence R. Rabiner “A tutorial on hidden Markov models and selected applications in speech recognition”, Proceedings of the IEEE 77.2, pp. 257-286, 1989.
15. Bilmes, Jeff A. "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models." *International Computer Science Institute* 4.510 (1998): 126.
16. Viterbi, Andrew. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm." *IEEE transactions on Information Theory* 13.2 (1967): 260-269.
17. [4] A. G. Maher, R. W. Kind, J.G. Rathmell. A Comparison of Noise Reduction Techniques for Speech Recognition in Telecommunications Environments. In The Institution of Engineers Australia Communications Conference, Sydney, October, 1992.