Kungliga Tekniska Högskolan
Valhallavägen 79
100 44 Stockholm

Phoneme Recognition Using Deep Neural Networks

Course project(DT2118)

Authors
Balasubramanian Rajasekaran
Deepa Krishnamurthy

Teacher
Giampiero Salvi

Abstract

In recent years, Automatic Speech Recognition(ASR) System has gained a significant popularity in the field of Human Computer Interaction. Phonemes play an important role in Speech Recognition System as they are free from any vocabulary inhibitions. Thus, phoneme recognition is a fundamental aspect of ASR. In this project, a Deep Neural Network(DNN) is implemented and trained to recognize phonemes of digits 'zero' to 'nine'. Mel Frequency Cepstral Coefficients(MFCC), a widely known feature in ASR is used for training the DNN. The Neural Network is constructed, trained and tested with datasets using the Neural Network Toolbox of MATLAB 2015a.

1 Introduction

Speech is one of the most important communication methods that exist in today's world. The significance of this medium has led to the development of Automatic Speech Recognition (ASR) systems. Although ASR has made a significant progress in many applications like voice recognition and vocabulary transaction, its performance is yet to reach a level of a stabilized and powerful user interface. In this project, an approach to improve ASR's effectiveness to recognize phonemes is experimented using Deep Neural Networks (DNN).

Phonemes are the acoustical elements which forms the smallest meaningful distinguishable unit of speech. They are produced by glottis, lungs, vocal cavity, nasal cavity, tongue, and other body parts. Phonemes are combined together to form syllables which is in turn combined to form words. For instance, to pronounce the number 9, the phonemes used will be 'n', 'ay', 'n'. Thus, a phoneme recognition system is essential when using an entire vocabulary becomes challenging and cumbersome. Since the total number of phonemes for a given language is finite as shown in Figure 1, a phoneme recognizer should essentially identify phonemes given a set of speech features. Various feature extraction techniques exist such as Principal Component Analysis (PCA), Linear Discriminate Analysis (LDA), Independent Component Analysis (ICA), Linear Predictive Coding (LPC), Mel-Frequency Cepstral Coefficients (MFCC) etc. In this project, MFCC is used for feature extraction as this is the most widely used method in ASR.

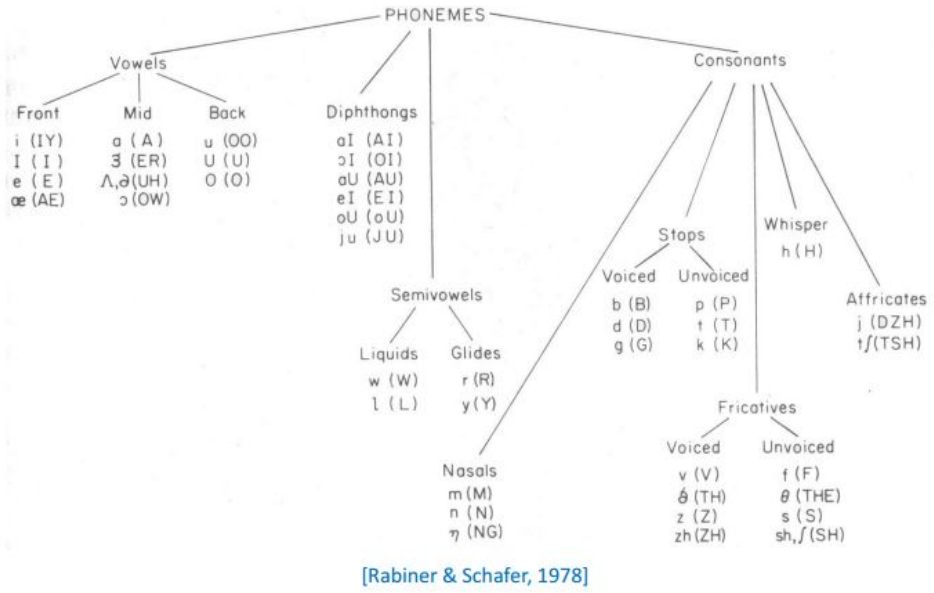


Figure 1: Phonemes Chart

There are several techniques that exist for modeling an ASR like Hidden Markov Model(HMM), Dynamic Time Warping(DTW), etc. However, the accuracy and performance capabilities of this system is still a concern. Hence in this project , DNN is used to improve the performance of an ASR in recognizing phonemes. A DNN is a neural network with many hidden layers between the inputs and outputs. MATLAB 2015a has released a toolbox for DNN computation and this is used for training and testing the MFCC datasets.

2 Related Work

There has been several work on Phoneme Recognition Systems. Many methods already exist for phoneme based Speech Recognition System such as Hidden Markov Models, Neural Networks, Gaussian mixtures for phoneme models and so on. In present day, there has been several techniques which clearly outperforms the traditional HMM and Gaussian mixtures. For instance, the techinque presented by this paper [4] performs better than HMM in Phoneme Recognition. In this paper, Large Reservoir Computing was used on TIMIT corpus. Large reservoirs of atleast 20000 nodes were configured in a hierarchial way. With 2 layers , 20K reservoir systems , competitive error rates were achieved as compared to the Hidden Markov Models. However, it is acknowledged that other techniques such as Deep Belief Networks are still better than reservoir computing.

Using Neural Networks for Phoneme Recognition as mentioned previously , has proven to be a good choice. For instance, in this paper [3] on Deep Belief Networks(DBN) for phone recognition, the back-propagation DBN and the associative memory DBN architecture were investigated. It was found that both architectures avoided overfitting. With changing the parameters such as number of hidden layers and the number of units per layer, the Phone error rates(PER) was 23% which is lesser than the PER of techinques such as CD-HMM, Recurrent Neural Nets and so on.

Another paper [5] on phoneme recognition with Time Delay Neural Networks had a significant reduction in error rates. Here, a 3-layer neural network is used with some delays added to the network. 16 melscale spectral coefficients serve as input to the network. This is interconnected to 2 hidden layers. A Back-propagation learning procedure was used in which sequences of patterns where used to achieve a translation invariant network. It was proven that TDNN yielded a considerable performance improvements over HMM with the error rate reducing from 6.3% to 1.5%. Besides this, TDNN was able to invent meaningful linguistic abstractions in time and frequency such as format tracking and segmentation.

3 Method

In this project, a deep neural network(DNN) approach was used to train and test the MFCC's to yield higher performance rates and accuracy. A DNN has multiple layers of neurons - input layer, hidden layers and output layer as shown in Figure 2.

The DNN in the project is constructed using the Neural Network Toolbox of MATLAB 2015a. Once the network is created, the weights and bias of the links between the layers has to be initialized. It can be done by sampling a distribution or by unsupervised pre-training methods. Pre-training is performed because, choosing the initial weights that approximates the final solution can improve learning [1]. One of the techniques to do pre-training is by using Auto-Encoders.

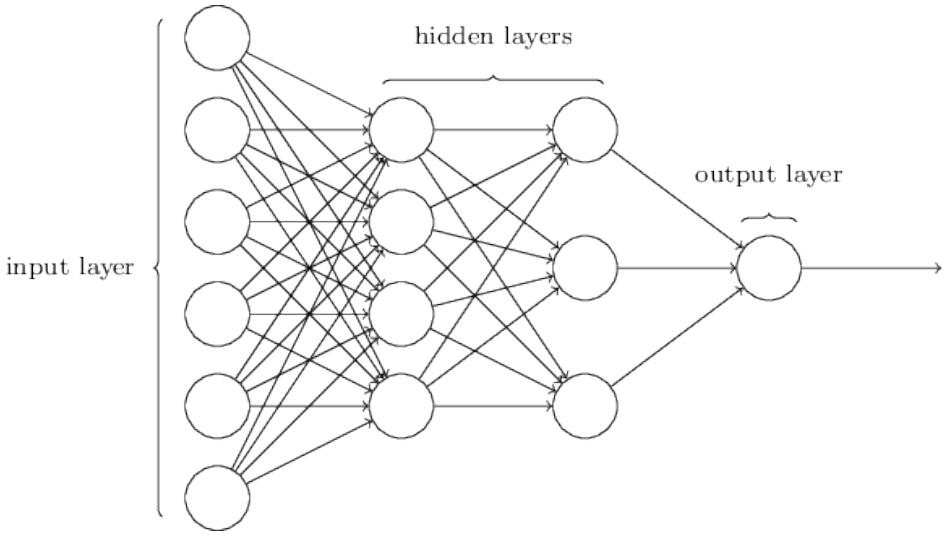


Figure 2: A Deep Neural Network with 2 hidden layers

3.1 Auto-Encoders

This is a pre-training technique developed by Geoffrey Hinton for training many-layered "deep" network treating each neighboring set of two layers like a Restricted Boltzmann Machine (RBM). pre-training to approximate a good solution and then perform a supervised fine-tuning with back-propagation technique [2].

Instead of training it to predict some target value y given inputs x, an auto-encoder is trained to reconstruct its own inputs x as shown in Figure 3.

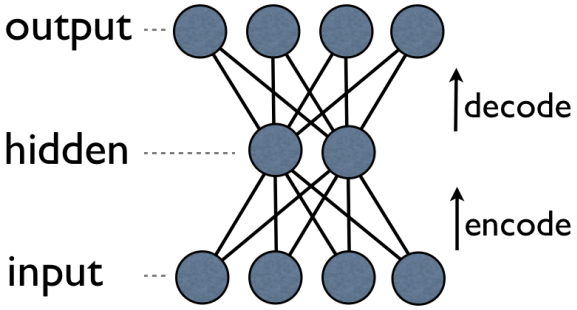


Figure 3: Auto-encoding network

4 Experiments

4.1 Data Extraction

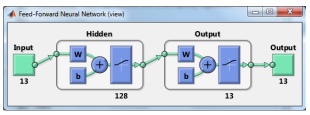
A huge amount of data is required to train a DNN. For this project, the first 13 coefficients of the MFCC's of the various phonemes in the digits -'zero' to 'nine' are used to train the DNN. The MFCC's were extracted from the speech files of the tidigits dataset by parsing the result of HTK alignment tool and annotating the MFCC of the speech into its corresponding phonemes. The dataset which is collected is stored as a '.mat' file in the format {X,Y} where X is the MFCC and Y is the corresponding phoneme label. The dataset contains multiple MFCC's for the same phoneme.

4.2 Training and Testing using DNN

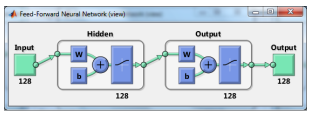
As shown in the Figure 5, the DNN created using Neural Network Toolbox of MATLAB 2015a has 2 hidden layers containing 128 neurons each and the output layer consisting of 22 neurons(each corresponds to a phoneme). The DNN is trained by using the data randomly picked from the dataset such that the number of data chosen for each phoneme is kept constant.

The trained network is tested with a set of 10 new MFCC data per phoneme and checked if the output of the network corresponds to its phonemes.

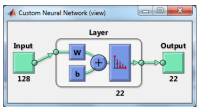
The system was experimented by training the DNN with different amount of training data, different number of nodes in the hidden layers and for different epochs for the auto-encoders in Figure 4 and the network.



(a) I/P layer and first hidden layer



(b) First hidden layer and second hidden layer



(c) Second hidden layer and O/P layer - Back propagation

Figure 4: Auto-encoders for the network

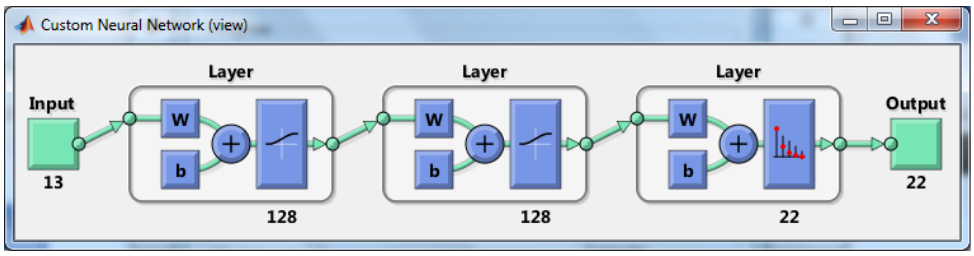


Figure 5: The complete Neural Network with 2 hidden layers and 128 nodes for each hidden layers as shown in the MATLAB 2015a toolbox.

5 Results

Having experimented with the size of the dataset used for training, number of nodes in hidden layers and epochs it was found that with 8000 MFCC data per phoneme i.e 8000 x 22 = 176000 data in total, 128 nodes per hidden layer, 500 epochs for auto-encoders and 1000 epochs for the complete neural network gave the highest accuracy of 70.5% when compared to using 2000 MFCC data per phoneme or using 100 epochs for auto-encoders and 500 epochs for the complete network which gave an accuracy of 59.5%.

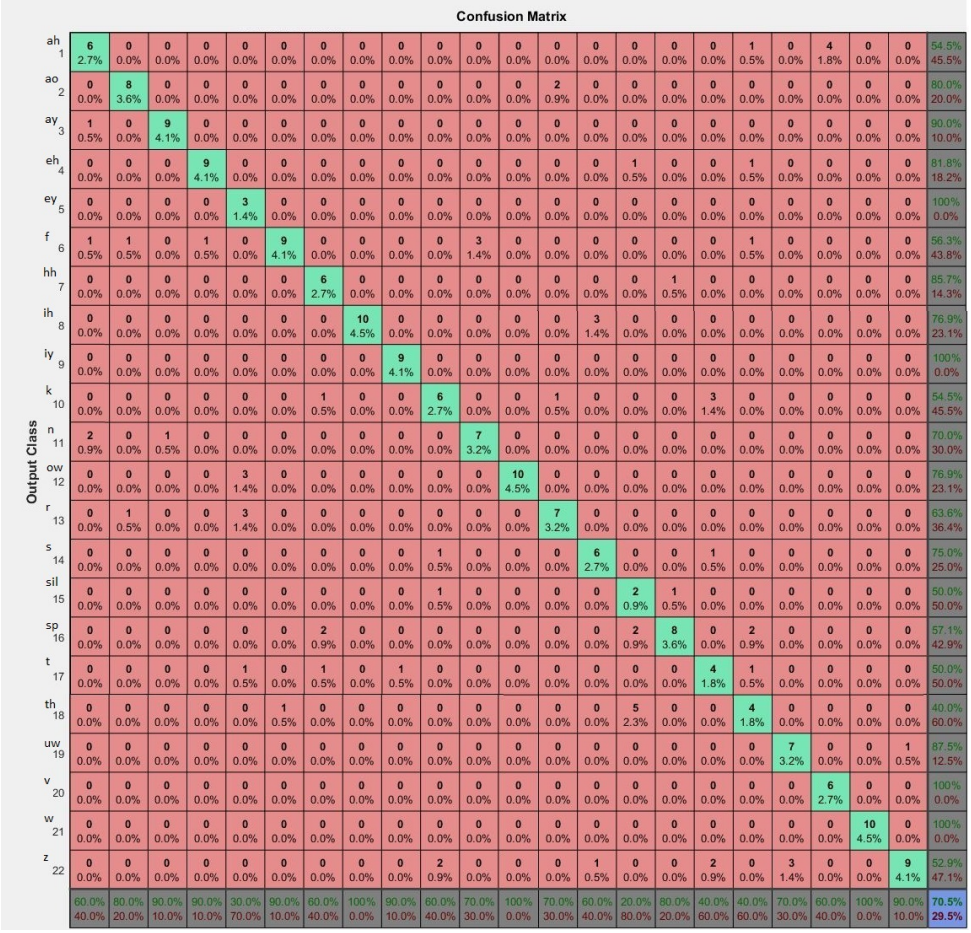


Figure 6: Confusion Matrix for the above experiment conducted

6 Conclusion and Discussion

It can be concluded that a considerable amount of accuracy was achieved using 2 hidden layers and MATLAB Neural Network Toolbox. From the Figure 6, it can be observed that few phonemes like 'ow', 'ih' and 'w' yielded 100% accuracy while 'sil' yielded 20%. This is because the silence region may contain some noise/energy regions due to background disturbances.

Due to the hardware constraints, the number of layers was limited to 2 but with the help of GPU this can be extended to many more layers. The implementation of DNN could also be done in less heavy framework like Python or C++ which will be computationally more efficient. MATLAB Neural Network Toolbox was used to experiment and explore the features and options that it provides.

It is also worth mentioning that by using LPC data, the accuracy can be boosted to almost 98%. Thus, these suggestions could help new experiments recognize phonemes in a better way.

References

[1] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, 11:625–660, 2010.

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

[3] Abdel-rahman Mohamed, Tara N Sainath, George Dahl, Bhuvana Ramabhadran, Geoffrey E Hinton, and Michael A Picheny. Deep belief networks using discriminative features for phone recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5060–5063. IEEE, 2011.

[4] Fabian Triefenbach, Azarakhsh Jalalvand, Benjamin Schrauwen, and Jean-Pierre Martens. Phoneme recognition with large hierarchical reservoirs. In *Advances in neural information processing systems*, pages 2307–2315, 2010.

[5] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(3):328–339, 1989.