Deep Belief Networks for Kannada Phoneme Recognition

Akhila K.S
M.Tech Student, Signal Processing
Siddaganga Institute Of Technology
Tumakuru,Karnataka,India

R.Kumaraswamy, PhD
Professor & HOD, Dept. of ECE
Siddaganga Institute Of Technology
Tumakuru, Karnataka, India

ABSTRACT
In this paper, a baseline phoneme recognition system for Kannada language is built using MFCC and Deep Belief Networks (DBNs). Phonemes are segmented from continuous Kannada speech and MFCC features are extracted from each speech frame. These features are further used as input to the recognizer. DBNs are probabilistic generative model which are constructed by stacking Restricted Boltzmann machines (RBMs). The learning procedure of DBN undergoes pre-training phase followed by fine-tuning phase. Evaluations are also carried out on conventional speech recognition methods such as Multi-Layer Feed Forward Neural Networks (ML-FFNNs) and Support Vector Machines (SVMs). The Experimental result shows that DBN's performance is superior to the conventional methods for recognition of Kannada phonemes using MFCC features.

Keywords
Kannada phoneme recognition, MFCC features, Deep Belief Networks (DBNs), Multi-Layer Feed Forward Neural Networks (ML-FFNNs), and Support Vector Machines (SVMs).

1. INTRODUCTION
Automatic Speech Recognition (ASR) has evolved significantly with the advent of new machine learning algorithms. One of the interesting application areas of ASR is audio mining, which is used to search audio or video files for occurrence of spoken words or phrases. The speech recognition search engine identifies words or phonemes that are spoken within the audio or video files and generate a searchable index that includes a time stamp for each important words or phoneme and its locations within the file [1, 2]. Thus it can enable users to search audio and video files similar to the text search engine [3,4].

Kannada language is widely spoken in the state of Karnataka. Hence it would be beneficial to build phoneme based search engine for Kannada language. Many research works has been done on regional languages such as Kannada, Malayalam, Telugu, Hindi and Punjabi [5, 6, 7]. But only few research works have contributed to Kannada language. The proposed work in this paper is a step towards developing speech recognition system for Kannada language.

Machine learning technique is one of the most active research areas of speech recognition. There are different approaches to the speech recognition, each method has its advantages and limitations. Typical ASR system uses statistical pattern recognition framework called HMM/GMM (Hidden Markov Model/Gaussian Mixture Model) which is a generative model [8, 9]. The hidden states of HMM/GMM has limited representational capacity and makes unrealistic independence assumptions. If the target of ASR is pattern classification, discriminative models may be an appropriate choice for acoustic modeling than the generative models. Neural Networks (NNs) are traditionally been trained discriminatively. ML-FFNNs adopting error back-propagation can learn more complicated functions [11]. Unfortunately this method did not work well as back-propagation is based on local gradient descent, and which usually starts at some random initial weights. It often be trapped in poor local optima, and severity increases significantly with increase in depth of the network. This difficulty is partially responsible for steering the research on machine learning from neural networks to SVMs, for which global optimum can be efficiently obtained. Despite the fact that SVM can work well in many practical problems [12,13], encounter its constriction due to shallow architectures.

To overcome the limitations of conventional classification methods, Hinton et. al., [14] introduced an unsupervised learning procedure called Deep Belief Networks (DBNs). DBN is a hybrid model consisting of multiple layers of Restricted Boltzmann Machine (RBM). The greedy layer-by-layer training is employed to efficiently learn a deep and hierarchical probabilistic model. This learning procedure can optimize the network weights so as to provide better initialization to MLPs as compared to the random weights.

This paper aims to build a baseline phoneme recognition system for Kannada language using Mel-Frequency Cepstral Coefficients (MFCCs) and Deep Belief Networks (DBNs). Also the results of other two classifiers such as ML-FFNNs with error back-propagation learning and SVMs are compared with DBNs. The organization of the paper is as follows, Kannada phonemes and collection of database are explained in section 2. Section 3 and 4 describes the methodologies of ASR, feature extraction and review of discriminative classification methods respectively. Section 5 and 6 summarizes the results and conclusion.

2. KANNADA LANGUAGE
Kannada language has forty nine phonemes, which are divided into three groups: Swaragalu (13 letters-vowels); Yogavaahakagalu (2 letters); and Vyanjanagalu (34 consonant phonemes which are divided into voiceless/ voiceless aspirated consonants, voiced/voiced aspirated consonants, unstructured consonants), similar to English vowels and consonants, respectively. Corresponding to syllables, Kannada language is composed of akshara or kagunitha. Most of the syllables have the following structure (vowel-V & consonant-C) V, VC, CV, CVC and CVV. The ligaturing rules govern the large set of symbols in an akshara system (e.g.:ಕ್ಷ, ಕ್ಷ್ಮ). Phoneme recognition in continuous speech is a difficult task with a low accuracy rate due to the following reasons:

1. Complex structure of vowels and consonants, hence there is difficulty in deciding which phonemes are being occurred.

2. Highly confusable acoustic similarities of phonemes.
3. The phonemes are short in time with low energy.

Each unique phones of a language are assigned an International Phonetic Alphabet (IPA) symbols. In this work, 41 phonemes are considered, perceptually similar and least occurring phonemes are merged to form 25 phonemes. The phoneme merging information is mentioned in Table 1. Broadcast/Read mode of Kannada speech is used in order to get a series of phoneme. An example of Kannada read mode speech transcription is illustrated in Figure 1.

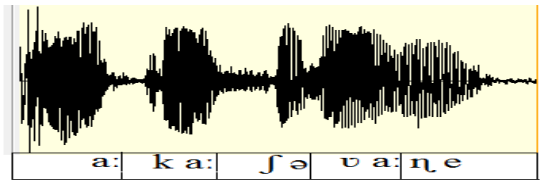

Fig. 1: An Example of Kannada continuous speech and its transcription

3. SPEECH RECOGNITION METHODS

In this section, the methodologies of ASR are discussed. Three discriminative classification methods viz., ML-FFNNs with back-propagation learning rule, Support Vector Machines (SVMs), and Deep Neural Networks (DBNs) are mentioned.

3.1 Multi-Layer Feed Forward Neural Networks (ML-FFNNs)

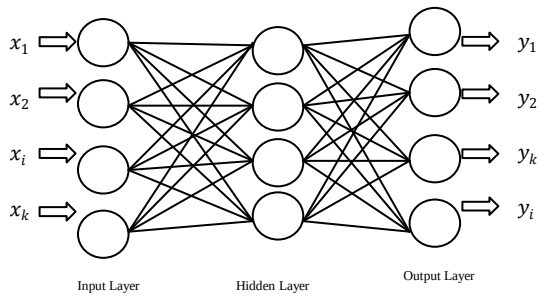

Fig. 2: Multi-layer Feed Forward Neural Network

In this work, the conventional method of classification using ML-FFNNs with back-propagation learning algorithm is used as one of the method for comparison. In back-propagation learning algorithm error signal computed at the output layer gets back-propagated to get derivatives, in order to update the weights until convergence is reached. Figure 2 shows the architecture of a Multi-layer Feed Forward Neural Networks with error back-propagation.

A fixed dimension speech features are given at the input layer and its nodes are set from the length of feature vector. The number of hidden layers and nodes are empirically selected. The output layer nodes are decided based on number of classes. To represent each class a target labels are assigned at the output layer.

Table 1.Phoneme Merging Table

S.l.No.	Kannada	Phonetic Unit(in IPA)	Merged Phonetic Unit(in IPA)	Corresponding name (in ASCII)
1	ಅ (short a)	ə	a	a
2	ಅ	aˑ	a	a
3	ಆ	aː	a	a
4	ಎ	eː	e	e
5	ಎ	e	e	e
6	ಇ	i	i	i
7	ಈ	iː	i	i
8	ಒ	o	o	o
9	ಓ	oː	o	o
10	ಉ	u	u	u
11	ಊ	uː	u	u
12	ಕ್	k	k	k
13	ಗ್	g	g	g
14	ಜ್	dʒ	dʒ	j
15	ಟ್	ʈ	ʈ	T
16	ಡ್	ɖ	ɖ	D
17	ತ್	t	t	t
18	ಐ	ŋ	ŋ	n
19	ಪ್	p	p	p
20	ಬ್	b	b	b
21	ಮ್	m	m	m
22	ಯ್	j	j	y
23	ಿ,ರ್ Short ra	ɾ	r	r
24	ರ	r	r	r
25	ಲ್	l	l	l
26	ವ್	ʋ	v	v
27	ಶ	ʃ	ʃ	sh
28	ಷ	ʂ	ʃ	sh
29	ಸ್	s	s	s
30	ಹ್	h	h	h
31	ಳ್	ɭ	ɭ	L
32	ಖ್	kʰ	k	k
33	ಘ್	gʱ	g	g
34	ಛ್	ʤʰ	ʤ	ch
35	ಝ್	dʒʱ	dʒ	j
36	ಠ್	ʈʰ	ʈ	T
37	ಢ್	ɖʱ	ɖ	d
38	ಥ್	tʰ	t	t
39	ಧ್	dʰ	D	D
40	ಫ್	pʰ	p	p
41	ಭ್	bʰ	b	b

The weights are initialized randomly and back-propagation learning algorithm is employed to update the weights during training with mean square error as given in equation (1)

$$E = \frac{1}{2}\sum_{l=1}^{L}[y_d(p) - y(p)]^2 \qquad (1)$$

Where $y_d(d), y(p)$ is the desired output and actual output respectively. Weights are updated through minimizing the error E. The weight update is done using equation (2).

$$W_{jk}(p+1) = W_{jk}(p) + \Delta W_{jk}(p) \quad (2)$$

Where W_{jk} is the weight connection between output layer and the hidden layer. The term $\Delta W_{jk}(p)$ defines the weight correction is as given by equation (3)

$$\Delta W_{jk}(p) = \eta y_j(p)\delta_k(p) + \alpha \Delta W_{jk}(p-1) \quad (3)$$

In which η is the learning rate, $\delta_k(p)$ is the error gradient at neuron k of iteration p and α is momentum which is a constant.

3.2 Support Vector Machines (SVMs)

SVM maps the input data into high-dimensional space. It tends to work well in many practical problems due to its simple architecture. The classification is based on the idea of decision hyper-planes which determines the boundaries in input space form a set of labeled training dataset. SVM is mainly a binary classification technique: hyper-plane will try to split the positive samples from the negative samples. Since real world problem deals with multi-class classification which can be solved using two approaches: One-Against-All (OAA) and One-Against-one (OAO) approach. In this paper, LIBSVM tool is used to perform classification using OAO approach, which constructs $\boldsymbol{M \times (M-1)/2}$ binary classifiers, using all the binary pair-wise combinations of the $\boldsymbol{M}$ classes. Considering each class that can be qualified by using the example of the first class as positive and the second class as negative, to combine these classifiers the Max Wins algorithm is used. It finds the subsequent class by selecting the class voted by the majority of the classifiers.

3.3 Deep Belief Networks (DBNs)

DBNs are probabilistic generative models, constructed by stacking Restricted Boltzmann Machine (RBM) [15, 16,17] as illustrated in the Figure 3. The learning procedure of DBNs undergoes two steps: Pre-training phase and Fine-tuning phase. The pre-training procedure of DBN is used to initialize the weights of MLP, which can be discriminatively fine-tuned by error back-propagation derivatives. An obtained DBN weights become the weights of standard ML-FFNNs.

3.3.1 Pre-training phase

DBN is treated as a stack of RBMs.

1. Restricted Boltzmann Machines

RBM is an undirected structure with symmetric weights connecting between units of visible layer and hidden layer. There is no connection between visible to visible or hidden to hidden, and hence the term "restricted" is used.There are two types of RBM depending on the type of input data that each layer receives. If visible layer deals with real-valued input data and the hidden layer carry binary representations then RBM is of the form Gaussian-Bernoulli. If both layer of RBM carries binary representation, it is said to be of the form Bernoulli-Bernoulli RBM.

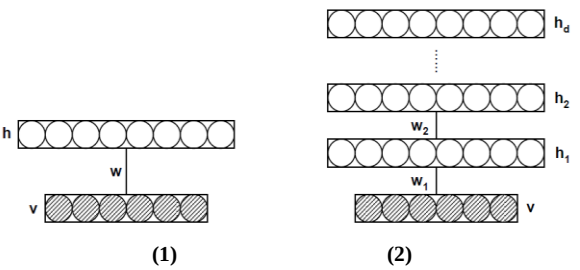

Fig. 3: (1) RBM and (2) DBN-Stacking RBM

Typically, all visible units are connected to all units of hidden layer. Let v be the input feature or visible vector and h represents hidden layer.

$$p(v,h;\theta) = \frac{\exp(-E(v,h;\theta))}{Z} \quad (4)$$

Where Z is the Normalizing Constant as given in equation (5)

$$Z = \sum_v \sum_h \exp\big(-E(v,h;\theta)\big) \quad (5)$$

Equation (6) gives marginal probability of model declared to a visible vector$v$.

$$p(v;\theta) = \frac{\sum_h \exp(-E(v,h;\theta))}{Z} \quad (6)$$

If visible and hidden units are Bernoulli stochastic units, energy function and the conditional distribution is obtained from (7) and (8,9) respectively.

$$E(v,h;\theta) = -\sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij}\, v_i h_j - \sum_{i=1}^{V} b_i\, v_i - \sum_{j=1}^{H} a_j\, h_j \quad (7)$$

Where the parameter are defined by the model,$\theta = \{w, b, a\}$, w_{ij} is the weights between i^{th} unit of visible layer and jth unit of hidden layer, b_i&a_j are biases for i^{th}visible unit and jth hidden unit respectively, V and $H$are the number of visible and hidden units respectively.

$$p\big(h_j = 1 \big| v;\theta\big) = \sigma\left(\sum_{j=1}^{V} w_{ij}\, v_i + a_j\right) \quad (8)$$

$$p\big(v_j = 1 \big| h;\theta\big) = \sigma\left(\sum_{j=1}^{H} w_{ij}\, h_j + b_i\right) \quad (9)$$

For Gaussian-Bernoulli stochastic units, the energy function and conditional probabilities of jthhidden andi^{th}visible units are given by (10) and (11,12) respectively.

$$E(v,h;\theta) = -\sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij}\, v_i h_j + \frac{1}{2}\sum_{i=1}^{V}(v_i - b_i)^2 - \sum_{j=1}^{H} a_j\, h_j \quad (10)$$

$$p\big(h_j = 1 \big| v;\theta\big) = \sigma\left(\sum_{j=1}^{V} w_{ij}\, v_i + a_j\right) \quad (11)$$

$$p(v_i = 1|h;\theta) = N\left(\sum_{j=1}^{H} w_{ij}\, h_j + b_i\right) \quad (12)$$

1. Contrastive Divergence (CD)

Contrastive Divergence procedure-weight update rule for training RBM following the log likelihood gradient, $\log\, p(v;\theta)$. The change in gradient is given by equation (13).

$$\Delta w_{ij} = \eta\big(< v_i h_j >_{data} - < v_i h_j >_{reconstruction}\big) \quad (13)$$

Where η is the learning rate, $< v_i h_j >_{data}$ is the inner product term observed in the training set and $< v_i h_j >_{reconstruction}$ is the expectation under distribution defined by the model.

For multi-layer RBM, Greedy layer-wise training procedure is followed [14]. In which RBMs are stacked to form DBN. Initially Gaussian-Bernoulli RBM is learnt, the activations from the hidden unit are considered as input to the Bernoulli-Bernoulli RBM. The hidden unit activations of the second layer are treated as input to the preceding layers of RBM, and so on. Maximum likelihood learning can be achieved from greedy layer-by-layer RBM stacking [15, 16, 17].

3.3.2 Fine-tuning phase

The pre-trained weights obtained are used as the initialization for conventional ML-FFNNs which provides better initialization as compared to random weights. The weights are fine-tuned using Error Back-propagation learning algorithm [18, 19].

4. EXPERIMENTAL SETUP

The Experimental setup and block diagram of the proposed phoneme recognition method is mention in this section.

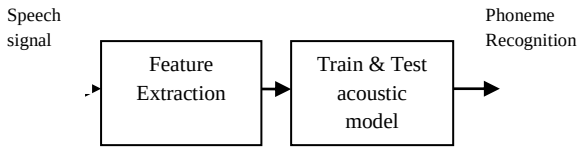

Fig. 4: Block diagram of the proposed work

4.1 Database

Database consists of Kannada phonemes which are extracted from broadcast/read mode speech. For classification 40 utterances per phoneme are used for training and 10 utterances per phoneme are used for testing.

4.2 Feature Extraction

For any speech task MFCCs are most widely used spectral features. Shape and size of the vocal tract are changed for different phonemes being produced. This shape can be represented through an envelope of short term power spectrum. MFCCs approximately represents this envelop. In this work, 16 MFCCs along with short time energy are extracted from each frame. To eliminate the variability in the input speech, obtained feature vector is normalized to zero mean and unit variance. Table 2 shows the parameters used for feature extraction.

Table 2. Parameter selection for feature extraction

Parameters	Value
Sampling rate	8KHz
Pre-emphasis co-efficient	0.97
Window type	Hamming
Frame size	25ms
Frame shift	10ms

4.3 Parameter Selection

4.3.1 For MLPs with back-propagation

Input layer receives MFCC feature as a column vector of length 300. The hidden layers and their nodes are empirically chosen. Two hidden layers are considered with 140 and 80 nodes at first and second hidden layer respectively. The output layer nodes are set to 25 in order to recognize all the 25 phonemes. The learning rate is set to 0.01. The sigmoid activation function is used at hidden and output layer.

4.3.2 SVMs setup

25 phonemes are considered for classifications. Hence it has 25 target labels each target corresponds to each phoneme which results in 25*(25-1)/2=300 binary SVMs. Each one needs to be constructed such that it achieves optimal generalization performance. The Radial Basis Function (RBF) kernel is considered.

4.3.3 DBNs setup

The training configuration for RBM is given in Table 3 which is based on the practical guide of training RBM by Hinton et. al [17]. Two hidden layers of 1025 hidden units are used. By this the advantage of deep architecture is utilized [14].

Table 3. RBM Training configuration

Parameter	Value
Learning rate	0.001
Momentum	0.5 (1^{st} 10 epochs) and 0.9 (rest of the epochs)
Total epochs	200
Mini-batch size	20
Initial weights	N(0,0.01)
Initial visible bias	0
Initial hidden bias	-2
Evaluation criteria	Mean Square Error

5. EVALUATIONS

Three experiments are conducted. In the first experiment, conventional ML-FFNNs with two hidden layer are evaluated using back-propagation learning algorithm. In the second experiment, phoneme classification is performed using SVM with RBF kernel. DBN is used for classification in the third experiment. For DBN an unsupervised contrastive divergence procedure is used for pre-training. These pre-trained weights are fine-tuned using error back-propagation learning algorithm. Using three experiments, Table 4 provides the recognition accuracies for 25 Kannada phonemes classification and Table 5 provides the recognition accuracies for different phoneme categories viz., vowels (merged into 5 phoneme classes) and consonants (voiceless/voiceless aspirated (5 phoneme classes), voiced/voiced aspirated (5 phoneme classes) and unstructured consonants (8 phoneme classes)).

Table 4. Recognition results for 25 Kannada phonemes in percentage (%)

Features	MLP with Back-propagation learning	Support vector machine	Deep Belief Networks
LPC	67.2	69.1	72.3
MFCC	68.9	74	76.4

The recognition accuracy is calculated as the ratio of the sum of all phonemes with correct recognition to the entire phonemes in testing set. Results show that the performance of DBN is high as compared with other two methods. It is because an unsupervised pre-training procedure puts the

parameter values in an appropriate range for further supervised training.

The recognition rate of the network is affected by the size of phonemes used in the training and testing set. As the number of phoneme increases, the performance decreases rapidly. Corresponding to the Table 4 and 5, the results for DBN with 5 vowel phoneme classes the recognition rate is 90.3 %. This is decreased to 82% for 8unstructured consonant phoneme classes and further deteriorates to 76.4 % for 25 Kannada phoneme classes.

Table 5. Recognition results for different phoneme categories in percentage (%)

	ML-FFNNs		**SVMs**		**DBNs**	
	LPC	MFCC	LPC	MFCC	LPC	MFCC
Vowels (5 phoneme classes)	62	82	70	84	73	90.3
Voiceless/voiceless aspirated consonants (5)	60	78	64	86	72	86
Voiced/voiced aspirated consonants (5)	68	76	72	82	80	86
Unstructured consonants (8)	60	77.5	61	80	67.5	82

6. CONCLUSION

A baseline Kannada phoneme recognition system is built using MFCC features and DBNs. The performance of DBN is compared with the conventional methods of speech recognition such as, ML-FFNNs and SVMs. The ML-FFNNs with back-propagation learning are often stuck in poor local minima due to random weight initialization. SVMs overcome the problem of local minima, due to its shallow architecture which lacks the ability of learning features. DBN is an efficient method for learning feature with deep architecture. Results highlight the superiority of DBN when compared with ML-FFNNs and SVMs using MFCC features.

7. SCOPE FOR FUTURE WORK

DBNs are new area of pattern recognition. Hybrid architecture can be formed using HMM-DBN for significant improvement in the performance of speech recognition by exploiting the advantage of both generative and discriminative acoustic models.

8. REFERENCES

[1] D.H.Klatt, "Overview of the ARPA speech understanding project". In Lea, W.es.Trends in Speech Recognition, Englewood Cliffs, NJ: Prentice-Hall. 1980.

[2] K.Ng and V.Zue. "Phonetic Recognition for Spoken Document Retrieval", In Proceedings of ICASSP 98, pp.325-328.1993.

[3] Clements, Mark, P. Cardillo and Michael Miller, "Phonetic searching of digital audio", Proceedings, conference of the National Association of Broadcasters. 2001.

[4] J.R.Rohlicek, P.Jeanrenaud, K. Ng, H. Gish, B. Musicus, M. Siu, "Phonetic training and language modeling for word spotting", ICASSP, 1993.

[5] P. Saini, P Kaur and Mohit Dua, "Hindi Automatic Speech Recognition Using HTK", International Journal of Engineering Trends and Technology (IJETT)- Volume 4 Issue6- June 2013

[6] M. Dua, R.K.Aggarwal, V Kadyan and Shelza Dua, "Punjabi Automatic Speech Recognition Using HTK", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, July 2012.

[7] Shridhar M.V, Bapu K. Banahatti, Narthan L, Veena Karjigi, R. Kumaraswamy, "Development of Kannada Speech Corpus for Prosodically Guided Phonetic Engine",Oriental COCOSDA ,2013.

[8] M. Gales and S. Young, " The Application of Hidden markov Models in Speech Recognition ", Foundations &Trends in Signal Processing, vol, 1,no. 3, pp.195-304, 2007.

[9] Matthew Nicholas Stuttle, "A Gaussian Mixture Model Spectral Representation for Speech Recognition", Hughes Hall and Cambridge Univ. Engg. Dept.,July 2003.

[10] Chau Giang Le, "A thesis on Application of back-propagation neural network for Isolated Word Specch Recognition", Naval PG school, Monterey, California, June-1993.

[11] Paulraj M P, Sazali Bin Yaacob, Ahamad Nazriand Satheesh Kumar, "Classification of Vowel Sounds Using MFCC and Feed Forward Neural Network", International Colloquium on Signal Processing and Its Application , pp 59-62, 2009.

[12] M.A. Al-Alaoui, L. Al-Kanj, J.Azar and E. Yaacoub, "Support Vector machine (SVM) for English handwritten Character Recognition", Second International Conference on Computer Engineering and Applications, IEEE DOI 10.1109/ICCEA. 2010.56, 2010.

[13] Fereshteh Falah Chamasemani, Yashwant Prasas Singh, Multi-class Support Vector Machine (SVM) classifiers, "An Application in Hypothyroid detection and Classification", IEEE DOI 10.1109/BIC-TA.2011.51, 2011

[14] Mohamed, A.R. Dahl, G. E, and Hinton. G, "Acoustic Modeling using Deep Belief Networks", submitted to IEEE Trans on Audio, Speech and Language processing, 2010.

[15] A. Mohamed, G. Dahl and G. Hinton, "Deep Belief Networks for Phone Recognition", in Proc. of NIPS 2009 workshop on Deep Learning for Speech Recognition and Related Applications, 2009.

[16] Hinton . G., Osindero.S and Teh. Y, "A fast learning algorithm for deep belief nets", Neural Computation, vo. 18,pp. 1527-1554, 2006.

[17] G. E. Hinton, "A practical guide to training restricted Boltzmann machine", Tech Rep. UTML TR 2010-003, Dept. Computer. Sci., Univ. Toronto, 2010.

[18] Y. Bengio, Learning Deep Architectures for Artificial Intelligence", Jr Foundation and Trends in Machine Learning, vol. 2,pp. 1-127, 2009.

[19] Abdel Rahaman Mohamed, G. Hinton, Gerald Penn, "Understanding How Deep Belief Networks Perform

Acoustic Modeling", IEEE International Conference on Digital Object Identifier, pp. 4273-4276, 2012.

[20] Pradeep. R and R. Kumaraswamy,"Comparison of conventional methods and deep belief networks for isolated word recognition", Proc. of IEEE National Conference on Communication, Signal processing and Networking (NCCSN), pp 1-5, 2014.

[21] L. Rabiner and B-H Juang, Fundamentals of Speech Recognition, Pearson Education India, 1st edition, 2008.