Machine Learning in Automatic Speech Recognition: A Survey

Article *in* IETE Technical Review · February 2015

DOI: 10.1080/02564602.2015.1010611

2 authors, including:

Jayashree Padmanabhan
Anna University, Chennai
57 PUBLICATIONS **326** CITATIONS

SEE PROFILE

IETE Technical Review
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/titr20

Machine Learning in Automatic Speech Recognition: A Survey

Jayashree Padmanabhan[a] & Melvin Jose Johnson Premkumar[b]
[a] Department of Computer Technology, MIT, Anna University, Chennai, India
[b] Department of Computer Science, Stanford University, Stanford, CA, USA
Published online: 23 Feb 2015.

Machine Learning in Automatic Speech Recognition: A Survey

Jayashree Padmanabhan[1] and Melvin Jose Johnson Premkumar[2]

[1]Department of Computer Technology, MIT, Anna University, Chennai, India, [2]Department of Computer Science, Stanford University, Stanford, CA, USA

ABSTRACT

Over the past few decades, there has been tremendous development in machine learning paradigms used in automatic speech recognition (ASR) for home automation to space exploration. Though commercial speech recognizers are available for certain well-defined applications like dictation and transcription, many issues in ASR like recognition in noisy environments, multilingual recognition, and multi-modal recognition are yet to be addressed effectively. A comprehensive review of common machine learning techniques like artificial neural networks, support vector machines, and Gaussian mixture models along with hidden Markov models employed in ASR is provided. A thorough review on the recent developments in deep learning which has provided significant improvements in ASR performance, along with its relevance in the future of ASR, is also presented.

***Keywords*:**
Automatic speech recognition, Gaussian mixture models, Hidden Markov models, Machine learning, Support vector machines.

1. INTRODUCTION

From the early part of the previous century, there has been curiosity in making computers do what only humans could perceive, like recognizing speech, understanding natural language, processing images, etc. Speech being the primary, most efficient mode of communication between human beings, research in speech recognition has received much enthusiasm for the past five decades right from the advent of artificial intelligence. Many reasons can be attributed to this enthusiasm ranging from mere technological curiosity to the desire of automating tasks using machines and providing more natural machine interfaces.

The study on speech analysis dates back to the beginning of the nineteenth century, when Homer Dudley of Bell Laboratories made the first proposal for a speech analysis and synthesis system in 1930s [1,2]. In 1952, an isolated digit recognizer for a single speaker was built by Davis et al. of Bell Laboratories [3], followed by a system that could recognize 10 syllables of a single speaker proposed by Olson and Belar et al. [4].

A significant achievement occurred in the year 1959 when a phoneme recognizer was developed to recognize four vowels and nine consonants utilizing statistical information about phoneme sequences in English [5]. This marked the first use of statistical syntax in speech recognition. A precious technique that becomes popular in the 1970s is dynamic programming for automatic speech recognition (ASR). A technique generally known as dynamic time warping was first suggested by Vintsyuk et al. [6]. At the same time, at Bell Laboratories, the focus was on the creation of an automatic speech transcription system, which is speaker independent and can handle the acoustic variability arising in speech from different speakers with varying regional accents [7]. This was to fulfil the goal of providing telecommunication services to the people, including voice dialling and command-based automation of phone calls. Another important technique in Bell's approach to ASR is the concept of keyword spotting which attempts to detect only prescribed words or phrases of particular significance in an utterance and neglects the other non-essential portions [8]. This is to accommodate speakers who often prefer to speak natural sentences rather than rigid common words. Conventional authentication mechanisms lose favour in security applications as biometric identification systems take on the lead. Voice is considered to be an important biometric, and multimodal recognition systems are springing up to improve the robustness of authenticity as discussed in [9]. Speech recognition can be extended to recognize speakers, exploiting the information present in the speech and various methods including exploiting from the excitation source are reviewed and presented

in [10,11]. The above-mentioned techniques had a profound impact on the advancements in ASR for the past three decades. In this paper, a review of the various machine learning (ML) techniques for ASR is presented.

The rest of the paper is organized as follows. Section 2 provides an overview of ASR and its architecture. Section 3 contains introduction to basic learning techniques in machine learning. Section 4 introduces machine learning in ASR followed by an overview of ANN(artificial neural network)/HMM (hidden Markov models)systems; a review on various SVM (support vector machine)/HMM based approaches for ASR and the fundamentals behind GMM (Gaussian mixture model)/HMM systems along with their advantages and disadvantages based on related works in literature. Finally, we present the recently introduced deep learning techniques that have been proven to significantly improve the ASR performance. Section 5 concludes the paper with a summary of the techniques discussed and an outlook to future research directions in the field.

2. SPEECH RECOGNITION: ARCHITECTURE AND MODELLING

The general architecture of a speech recognition system is given in Figure 1. Noise removal is an important pre-processing operation generally performed in any speech recognition system. There are multitudes of noise removal mechanisms that are proposed in the literature, and Singh et al. [12] have performed an analysis on some common speech enhancement techniques like windowing, wiener filtering, spectral subtraction, spectral amplitude estimation, etc. in subjective and objective manners.

The input speech signal is first passed to the auditory front end, which pre-processes the signal and produces spectral-like features. These features are then passed to a phone likelihood estimator that estimates the likelihood of each phone. The phone likelihoods along with the HMM model and the n-gram language model (LM) are used by the decoder to decode the speech. The output words are then sent to the parser to convert it into human readable form.

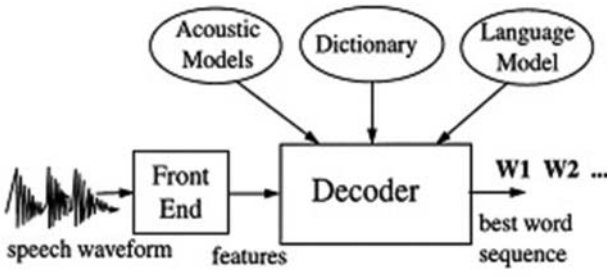

Figure 1: Speech recognition architecture.

2.1 Auditory Front End

This acoustic pre-processing unit aims at reducing the influence of undesired components in the speech utterance thus reducing the amount of training data. The typical sequence of steps involved in the front-end processing would be anti-aliasing filtering, analogue to digital conversion, windowing (usually hamming), fast Fourier transformation, computing power spectrum, and producing mel-frequency cepstral coefficients (MFCC) or any other coefficients like linear predictive, perceptual linear prediction, etc. This unit converts speech signal into speech frames and generates feature vectors, which describe the input speech signal.

2.2 Sub-Decoder

Speech recognition can be formulated statistically as follows. Given the acoustic observation $A = a_1, a_2, \ldots, a_k$, we have to find the word sequence $W = w_1, w_2, \ldots, w_n$ such that the probability $P(W|A)$ is maximized. Applying Bayes' rule we define the model as follows:

$$P(W/A) = \frac{P(A/W)P(W)}{P(A)} \tag{1}$$

where $P(A|W)$ refers to the acoustic model, $P(W)$ represents the LM, and $P(A)$ is a constant for a complete sentence.

2.3 Acoustic Model

This unit is used for extracting acoustic contents from the speech frames by modelling the acoustic input, using a sequence of states representing phone likelihoods. The acoustic model calculates the likelihood of the acoustic sentence A given W, the word sequence. By using GMM or a neural network to estimate the likelihood of each phone, along with the pronunciation lexicon, we could map words or could represent W in terms of the feature vectors, to model the durational and spectral variabilities of speech signals.

2.4 Language Model

The LM is the a-priori probability of observing the word sequence W, independent of the acoustic sequence A. Typically, n-gram LMs estimate the probability of a word w_k given the preceding $n-1$ words as

follows.

$$P(W) = \Pi(w_k \mid w_{k-1}, w_{k-2}) \quad (2)$$

3. MACHINE LEARNING

Over the years, sophisticated skills were developed to recognize patterns like speech, handwriting, facial features, etc. The pursuit to computer programs that make computers learn the above skills from the past experience gave birth to machine learning. Mitchell [13] stated in the context of machine learning that, "A computer program is said to learn from experience E with respect to some class of tasks in T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." Few related basic terminologies used with machine learning are introduced as follows:

- Example: an instance of the input.
- Features: an attribute set characterizing the input, represented as a vector or linear array.
- Labels: the category or class associated (e.g., positive or negative in binary classification or a real value in regression).
- Training data: data used to train the ML algorithm during learning phase.
- Test data: data used to test the performance of the learning algorithm during generalization phase.

Depending on how the machine gains knowledge to respond correctly, learning can be categorized into four basic methods as briefed in the following sections.

3.1 Supervised Learning

In supervised learning, the machine is trained with labelled data-set where output response or class for each input data vector is known. The assumption is that if the training data is large enough, a hypothesis that can perform well on the test data can be obtained. A simple example of supervised learning is curve-fitting problem. Given a set of input data, the machine is trained to generate the curved surface that best fits the training data-set, and during testing the machine is expected to correctly interpolate the new data over the curved surface. Feed-forward neural networks like perceptrons (adopting delta learning rule or perceptron learning rule), multilayer perceptrons (MLP, adopting back propagation), and constrained MLPs fall under this category.

3.2 Unsupervised Learning

In unsupervised learning, the machine is expected to learn the patterns in the unlabelled input data-set by itself without any feedback from the environment. The problem can be stated as finding the patterns in input data-set to partition or cluster the training data into subsets in an appropriate way. Taxonomic problems, where designing efficient ways to group the data into meaningful clusters, fall under this category. Examples are Hebb and Hopfield networks (Hebbian learning), Kohonen networks/self-organizing maps, and adaptive resonance theory (ART) networks/ART (competitive learning). Auto encoder is a simple network that is trained to produce what is given at the input, i.e. by setting the target output as the input. The network is trained to reproduce the input using gradient descent back propagation unsupervised learning method. Auto encoders are stacked to form a deep network that can be pre-trained using unsupervised learning to fix better initial weights and bias values.

3.3 Semi-supervised Learning

In semi-supervised learning, both labelled and unlabelled data are used for training the system. Typically, a small proportion of labelled data is used with a large amount of unlabelled data. This type of learning approach is usually adopted in problems where obtaining labelled data is very expensive.

3.4 Active Learning

In active learning, the algorithm interactively queries the user to obtain the labels for the examples. This is used in scenarios where unlabelled data is abundant but labelling the data is expensive.

4. MACHINE LEARNING IN ASR

ASR systems have already been deployed in many commercial applications but the problem of ASR is still largely unsolved. Over the years, various ML techniques have been employed for acoustic modelling in ASR systems. The reader is assumed to be familiar with Markov models used in prediction problems. For realistic problems, the outcome cannot be tied to a particular state of Markov model and can be estimated based on probability distribution associated with the states giving birth to HMMs. They are probably the most dominant techniques used for ASR. Ever since that introduction in ASR in the 1970s as evidenced from works in [14,15], they are considered to be the most significant paradigm shift in speech recognition [16]. Hence, HMMs can be considered to be the starting point of a speech recognizer. Three classical problems associated with HMMs are evaluation, decoding, and training. Given the model and observation sequence (outcome), determining the probability of the model in generating the sequence is evaluation,

being the *forward algorithm*; determining the most likely state sequence that generates the outcome is decoding, using the *Viterbi algorithm;* updating the model parameters for maximizing the likelihood of occurrence is training, adopted by *Baum-welch algorithm.*

However, HMMs have their own limitations, the most significant of which is the requirement of a large amount of training data to prevent the loss of performance due to the mismatch between the testing and training conditions. Typically, GMMs are used to estimate the output densities of these HMM state. These GMM/HMM systems are the most prominent generative learning approach used in ASR [17−20]. Nevertheless, ASR researchers never stopped considering alternate estimation approaches to be used with HMMs. This led to the exploration of various ANN-based approaches during the end of the 1980s and early 1990s. Numerous works in the literature have justified the use of ANNs or specifically multilayer perceptrons for probability estimation in ASR, as discussed in [21−23].

Another alternate probability estimation technique that has been investigated is the SVM. It should be noted that HMMs are generative models, i.e., the decisions are made based on the likelihood that the generative model has produced on the current pattern. However, since SVMs are discriminative in nature, they found favour among the research community in the mid-1990s. As generative and discriminative approaches are complementary, SVM/HMM hybrid systems were developed, much like the MLP/HMM systems, and have provided some interesting results [24,25]. SVMs have excellent generalization capabilities, which help to improve the robustness of ASR. This has led to the recent exploration of structured SVMs for noise-robust ASR systems [26].

4.1 Artificial Neural Networks for ASR

In this section, the role of ANNs in speech recognition is reviewed with the assumption that the readers are familiar with basic ANN architecture and basic learning functions. Readers may refer to [13] for basic concepts on ANN and other ML techniques. Over the past few decades, there have been various ANN-based approaches proposed aiming at overcoming the limitations of HMMs. We first look at ANN-based ASR systems wherein ANNs replace HMMs to model the time variability of the speech signal. Next, we glance at hybrid ANN/HMM systems where ANNs replace GMMs to model the probability densities. Finally, we give a glimpse of tandem ASR systems where tandem features are extracted using ANNs.

4.1.1 ANN-Based ASR Systems

A well-known way to attack the pattern recognition problem is to convert it into a spatial recognition problem wherein variants of multilayer, feed-forward neural network architecture are adopted to match the temporal structure of speech. Here, each speech unit is associated with a specific output unit in the MLP output layer [27,28]. Two commonly used types of ANNs are time-delay neural networks (TDNNs) [29] and recurrent neural networks [30−32]. Feedback is implemented by adding an additional vector containing the hidden unit values produced by the previous input. Another variant, where the output-layer loops back to the input layer, has been proposed in [33]. TDNNs, which are an alternative MLP architecture, have been investigated for deploying finite and infinite impulse response filters and for phoneme recognition in [29].

All the above models have been shown to perform on par (sometimes outperform HMMs) on isolated speech recognition involving short units. This is because these models require a target function to be defined, which is difficult if the training data consists of continuous speech where the segmentation is difficult. Hence, hybrid approaches that use MLPs to estimate the output probabilities of HMMs have become prominent. We review these hybrid approaches in the following section along with their strengths and weaknesses.

4.1.2 Hybrid ANN/HMM ASR Systems

The introduction of ANN/HMM hybrid systems was done by Bourlard et al. [21,22] in which an MLP was used to estimate the posterior probabilities of HMM states. The advantage of using feed-forward neural networks is the full utilization of the discriminative capabilities of ANNs and their ability to estimate the posterior probabilities as explained in [34] when trained by back propagation (BP) on an error criterion. In the BP algorithm, the gradient of a loss function with respect to all the weights is calculated. The gradient is then fed into an optimization method which updates the weights in order to minimize the loss function.

The idea was to use a standard HMM along with a neural network, where the later was used to estimate the state posterior probabilities. In order for the network to be trained by BP, it requires target values

for the output units to compute the cost function. However, supervised labelling of acoustic features is unavailable and it is not feasible to hand code for real-world tasks. To overcome this bottleneck, an iterative training procedure was proposed by Bourlard, which starts an initial segmentation of the acoustic features and then uses the Viterbi algorithm, along with the newly trained networks as probability estimators. This produces a more reliable segmentation of the initial data, which in turn can be used to train iteration in a similar fashion. A standard HMM was used to obtain the initial segmentation. An adoption of this approach was implemented in [35] on the resource management (RM) corpus, speaker-independent continuous task. Experimental results showed that on one of the test sets, the context-independent hybrid system (with 5.8% word error rate [WER]) outperformed the context-independent HMM (11.0%)

Many variations of the basic framework of Bourlard et al. have been proposed as narrated below. Radial basis function (RBF) [36] networks were used instead of MLPs in [37]. The resulting HMM/RBF hybrid system was used in an isolated word recognition task. In [38], the original framework was reinforced by generalizing the original connectionist probability estimates to global posterior of the models.

A context-dependent (CD) scheme to replace the context-independent HMM scheme was proposed in [39] by Franco et al. Experimental results on the RM SI task obtained 28% WER reduction compared to the context-independent system. In [40], a time window of previous acoustic features is given as input to the MLP along with the current acoustic feature vector, to take into account the correlation between the various acoustic vectors. Recurrent neural networks replaced MLPs as state-posterior estimators in a system called ABBOT developed by Robinson et al. [41]. The system used a combination of recurrent networks along with an efficient search space path pruning algorithm. A Viterbi alignment strategy is used on the output values of the second hidden layer of an ANN in [42]. This method is conceptually similar to that in [21] but does not explicitly use HMM as an estimator. Sequential MLP (SMLP) was proposed in [43]. Here, the outputs of the network are interpreted as discriminant functions that are able to discriminate between the various states. In [44], probability targets in the continuous range (0,1) were considered instead of the strict values of "0" and "1" used previously as the targets of the output units. For the task of recognizing continuous digits, results showed that this approach produced a WER of 4.9% whereas a standard system yielded a WER of 6%.

4.1.3 *ANN for Feature Generation*

Due to issues in learning MLPs, there has been subsequent research in the use of MLPs to generate bottleneck features [45]. These are features transformed from the standard acoustic units using a discriminative classifier. Typically, they are derived from the narrow middle layer of MLP. These features have been proved to possess high discriminative capabilities and are very robust against speaker and environmental vagaries.

Recently, many cross-lingual and multi-lingual experiments have been conducted with MLP-derived bottleneck features. In [46], an English-trained MLP feature generator improved the performance of Mandarin and Arabic ASR. Cross-lingual information was used in [47], wherein the connection weights of input-hidden layer and hidden biases of the Hungarian language MLP were initialized by the English-trained MLP weights, while the other weights were randomly initialized. Multilingual MLP features on five European languages, namely English, Italian, Spanish, Swiss French, and Swiss German, were used to classify context-independent phones in [48]. In [49,50], it was shown that the multilingual MLP was a good initialization scheme for MLP training in terms of both speech and accuracy through experiments conducted on both Vietnamese and Tamil languages. It was also shown that the multilingual MLP initialization scheme was very useful in rapid language adaptation to new under-resourced languages.

4.1.4 *Summary*

Although there has been a lot of work done on ANNs for ASR, they had soon fallen into disfavour due to some significant problems mentioned below, left to be addressed.

- Inability of ANNs to model time variability of the speech signal.
- Difficulty in designing optimal network architectures for hybrid models.
- Absence of a joint training scheme for training both HMMs and ANNs.
- Difficulty in learning large MLPs.

Recently, the difficulty associated with learning MLPs has been addressed. The development of new deep learning techniques has seen the reappearance of ANNs for large vocabulary continuous ASR. The problem of modelling temporal variability has also been addressed recently [51]. A discussion about these deep learning methods and other recent advancements is provided in the forthcoming section.

4.2 Support Vector Machines

4.2.1 SVM Definition

SVM is a binary non-linear classifier capable of predicting whether an input vector value x belongs to a class 1 (the desired output would be then $y = +1$) or to a class 2 ($y = -1$). Given a set of individualistic class data, the aim is to find an optimal decision parameter function. When the data is not linearly separable in the given space, it can be mapped to higher dimensional space and the "kernel trick" can be leveraged to perform efficient computation in the higher dimensional space. However, this still does not guarantee that the data will be linearly separable in the high-dimensional space. In [52], the idea of a soft-margin classifier was first proposed to make the algorithm work for non-linearly separable data-sets. The optimization problem now looks like as follows:

$$\min_{\gamma,w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i \tag{3}$$

s.t.

$$y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

where, for example, with margin less than 1, we pay the cost of the objective function being increased by ξ_i.

4.2.2 SVM for Pre-processing the Speech Feature Sequence

In most standard speech recognition systems, a three-state HMM is used to decompose the speech segment (phone or triphone) into a number of sections. The transition into and out of the segment is modelled by the first and third sections, whereas the second section models the stable portion. SVMs are used in many variants of this standard approach as referenced further. In [53], the authors show a significant performance improvement on a specific pattern classification task based on the Deterding vowel data and on a continuous alpha-digit one (OGI Alpha-digits) system. The vector obtained from the joint union of three segments from the triphone model is augmented along with the log of the time duration of the phone instance to explicitly model the variability of the system in duration. The composite feature vectors are derived from the alignments of a baseline three-state Gaussian-mixture HMM system. The SVM classifiers are then trained on these composite vectors, and the recognition process is also performed using these segment-level composite vectors. This approach has also been used on a large vocabulary conversational speech task (Switchboard).

A comparison between the system performance of classical HMMs and SVMs as sub-word units' recognizers for two different languages can be read in [54]. Forty-one monophone units are classified in a Japanese corpus and 86 consonant-vowel units are taken into consideration for an Indian language using two different strategies. For both Indian and the previously stated Japanese tasks, the SVMs have been proved to give a better performance than HMMs using the MFCC.

4.2.3 SVM for Continuous Speech Recognition

Continuous speech recognition is a more complex task when compared with isolated recognition, as it poses two major problems: temporal position of words or the number of words in the speech utterance is unknown and the size of the speech databases tends to be larger for continuous recognition tasks and, consequently, the size turns out to be larger than the maximum number of training examples, an SVM can handle. HMM/SVM systems for continuous recognition have been proposed analogous to HMM/MLP systems, where the phonetic level assignments generated by the HMMs are used by the SVM to classify the phonemes [24,25]. Since each segment may have a different duration, they need to be converted into fixed-length vectors using any one of the methods highlighted in the previous section. The authors suggested dividing the segment into three regions according to a pre-established proportion. Then, the vectors in every region were averaged and concatenated together. However, this method fails to exploit the generalization capabilities of SVMs. Moreover, the efficiency of the system is limited by the errors committed during the segmentation phase.

The proposal in [55] suggests classifying each voice frame as a phone. Thus, the need to locate the word in time and the duration becomes unnecessary. Token passing algorithm [56] is used for this and sequential minimal optimization (SMO) algorithm is used in [57] that allows for fast training of the SVM. The results obtained using these approaches are comparable to those obtained in HMMs. The computational complexity of the SVM classical formulation is addressed by using an alternative Lagrangian on the TIMIT database in [58].

4.2.4 Summary

SVM has its own advantages and disadvantages when it comes to ASR, though it is a state-of-the-art tool for classification problems. The improved discriminative capacity of SVMs lured many speech researchers to investigate them further. There are other advantages as well:

- SVMs are very robust and hence are apt for speech recognition in noisy environments.
- They are capable of dealing with inputs of thousands of dimensions, since only the kernel matrix is involved in the minimization process.

On the flip side, there are many disadvantages:

- Many SVM implementations require the storage of the entire kernel matrix of the input samples (n) in the memory. This has O (n^2) complexity and is one of the main drawbacks of SVMs.
- The output depends on the type of kernel used and there is no stipulation to determine which kernel is the best for a given task.
- The input vectors of an SVM need to be of fixed size, whereas in speech recognition each sequence can have variable duration.

4.3 Gaussian Mixture Models for ASR

GMMS are used for modelling continuous distribution components as parametric probability distributions (Gaussian or normal), and the entire data-set can be modelled using mixture of such distributions or Gaussians. GMMs are powerful in forming smooth approximations over a large class of sample distributions. GMM-based HMMs or GMM/HMM system is the most commonly used ML approach in ASR. A GMM/HMM system is represented by $\lambda = (\pi, A, B)$, where π is a vector of state prior probabilities; $A = (a_{i,j})$ is the state transition matrix; $B = \{b_1,\ldots,b_n\}$ is the set of GMMS of state j.

The HMM state is usually associated with a sub-segment of a phoneme in speech. A sentence is modelled by concatenating HMMs for the sequence of phones and GMM distribution is used to generate a vector in the HMM state. It can be represented as follows:

$$p(x/j) = \sum_{i=1}^{M_j} w_{ji} N\left(x : \mu_{ji}\sigma_j^i\right) \tag{4}$$

where N is the normal distribution with mean μ and variance σ, M_j is the number of sub-states for state j, and W_{ji} is the prior probability of sub-state i in state j.

4.3.1 GMM Formulation

The spectral features extracted from speech are real-valued but applying HMMs on continuous observations is not directly possible. Instead, the possible values of an observation feature vector o_t are assumed to be normally distributed. The observation likelihood function $b_j(o_t)$ is represented as a Gaussian. Given a data-set, the mean and variance can be obtained from the data, but the state that corresponds to an observation is not known. Hence, a way is needed to assign each observation vector o_t to every possible state i, incorporating the probability that the HMM was in state i at time t. Let this probability be $\Upsilon_t(i)$. Each vector of observation is modelled as a multivariate Gaussian with diagonal covariance matrices, and Baum–Welch algorithm is used to estimate the probability and to compute the mean and the variance.

A mixture of Gaussians is needed to model the multidimensional function and they need to be trained. The usual procedure to train the mixture of Gaussians (GMMs) is to choose M the number of Gaussians and splitting the Gaussian into two and running the forward–backward algorithm to retrain the Gaussians. This process is repeated until M Gaussians are generated.

Another approach is to do embedded training where each phone HMM embedded in an entire sentence is trained. Both word segmentation and alignment can be done as a part of the training process. Typically, CD phones are used and decision-tree-based state tying [59] is used to cluster the many states into various clusters.

4.3.2 Summary

There are few issues that need to be addressed when using GMMs.

- There is a need for variance flooring as this can improve generalization and prevent the variance from becoming very small.
- Using GMM increases computational complexity since a series of logarithmic additions is required to compute the GMM likelihood.
  - One way to deal with this is to include only components with a reasonable contribution to the total likelihood.
  - Another method would be approximating the likelihood by the maximum over all components.
- Determining the number of components per state in the system is another issue.
  - One approach is to use the same number for all states and determining that number with the help of data.
  - An alternative is to use the popular Bayesian information criteria. Another approach would be to design the number of components as a function of the number of observations in the state.
- Another well-known weakness of GMM is the conditional independence assumption.

On the other hand, the success and popularity of HMM/GMM systems are due to the following reasons:

- The main reason is the highly efficient Baum–Welch algorithm [60], which was the inspiration behind the general expectation−maximization (EM) algorithm [61], used for learning GMM and HMM models.
- The generative GMM/HMM has been found to successfully separate the noise from the speech in noisy speech utterances. It is interesting to note that they could exceed human performance when it comes to recognizing noisy speech. [62]

A recent advancement has been the introduction of subspace GMMs [63]. In this model, each speech state is a GMM, but the parameters of the GMM are not the parameters of the overall model. Instead, each state is associated with a vector-valued quantity of dimension similar to the feature dimension, and there is a globally shared mapping from this "state vector" to the means and weights of the state's GMM. This approach provides a much more compact representation and provides better results, especially on smaller amounts of training data.

4.4 Recent Advances in Machine Learning for ASR

In 2006, Hinton et al. proposed a novelty in learning [64,65], namely deep learning or hierarchical learning. Recently, these deep learning algorithms have been tested to show significant improvements in the performance of many problems including ASR. In deep learning, a hierarchical structure of processing layers is formed by exploiting unsupervised learning for pre-training of layers considering feature hierarchies and supervised back propagation for fine-tuning of pattern learning and classification. The recent popularity of deep learning can be attributed to the following two important reasons:

- Significant reduction in the cost of computing hardware.
- Drastic advancements in chip processing capabilities like graphics processing units (GPUs).

Since then, deep learning techniques have been successful in a number of diverse applications, namely natural language processing, computer vision, signal processing, voice search, audio processing, information retrieval, speech recognition and understanding, image processing, and robotics.

4.4.1 Introduction to Deep Learning

A well-known optimization bottleneck associated with deep models was alleviated when an efficient unsupervised learning algorithm was introduced in [65]. It led to the introduction of deep belief networks (DBNs), the main component of which is a greedy layer-by-layer learning algorithm that optimizes the DBN weights at a linear time. This is made possible with the help of restricted Boltzmann machines (RBMs), which are viewed as bipartite graphs with one layer of stochastic visible units (v) and another layer of hidden units (h) as shown in Figure 2.

The energy function $E(v,h)$ of an RBM is defined as follows:

$$E(v,h) = \sum W_{ij}.v_i.h_j + \sum b_i.v_i + \sum c_j.h_j \quad (5)$$

where W is a weight matrix connecting the two layers and b and c are the offsets in the visible and hidden layers, respectively.

DBNs can be perceived of layers having unsupervised pre-training of stacked RBMs followed by supervised fine-tuning. Since the publication of the seminal works in 2006, a number of variations have been proposed which are better and faster; being able to pre-train the DBN, layer by layer, considering the pairs of layers as a de-noising auto-encoder [66].

4.4.2 Deep Neural Network Formulation

A deep neural network (DNN) is nothing but a conventional MLP with many hidden layers. A DNN can be viewed as a directed graphical model that estimates the posterior probability $p(y = s|x)$ of a class S, given an observation vector x, as a stack of $(N + 1)$ layers of probabilities of hidden binary vectors h^n, given log-linear models. The first N layers model the posterior input vectors v^n. Each observation is propagated forward through the network, starting with the lowest layer. The output variables of each layer become the input variables of the next layer. In the last layer, the class posterior probabilities are computed as a multinomial distribution. To summarize, the estimation of $p(y = s|x)$ in the DNN consists of transforming the observation vector x into another feature vector using the multiple hidden layers. The transformed feature vector is then presented as input

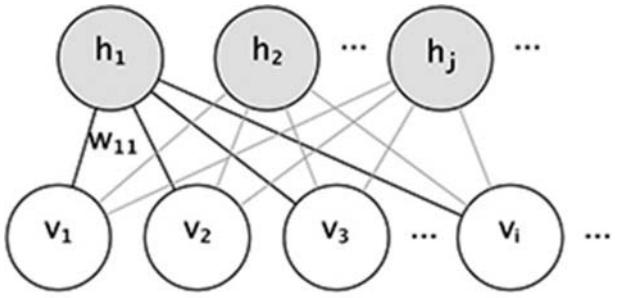

Figure 2: Graphical model of RBM.

into the log-linear model to estimate the posterior probability.

4.4.3 Deep Learning Techniques

Deep learning methodologies are classified into three types depending on what the networks are used for. The *generative deep architectures* characterize joint statistical distributions of data and their associated classes. Bayes' rule can be applied to derive equivalent discriminative architectures. Some examples are deep auto-encoders, deep RBMs, DBNs with Boltzmann machine as the base layer.

The second type consists of *discriminative models*, where characterizing posterior distributions of classes conditioned on the data often facilitates the pattern classification. Some examples are deep-structured conditional random field (CRF) which is a multi-layer CRF model where sequence training is employed [51], tandem MLP architecture [67], deep convex or stacking network [68] that are trained by stacking RBMs, and detection-based ASR architecture [69].

The third type is *hybrid deep architecture* which is a discriminative model utilizing a generative component to help in the discrimination. This can be done using better optimization and regularization or discriminative criteria are used to learn the parameters in any of the deep generative models as discussed in the first type.

4.4.4 Review

The first successful use of DNN-based acoustic models for large vocabulary continuous speech recognition (LVCSR) employed Bing voice search data [70]. It used five pre-trained layers of hidden units with 2048 units per layer and was trained to classify the central frame of an 11-frame acoustic context window using 761 possible CD states as targets. The system obtained a sentence accuracy of 69.6% on the test set, compared to 63.8% obtained by a GMM/HMM baseline. It was found that using tied triphone CD state targets was better than using monophone targets.

In [71], the above method was applied on the Switchboard speech corpus. Switchboard is a publicly available speech-to-text transcription benchmark task that allows much more rigorous comparisons among techniques. It consists of close to 300 hours of training data. The architecture used was a seven-layer DNN with 2048 neurons in each layer. Replacing the GMM with the DNN acoustic model reduced the WER from 27.4% to 18.5%. This is staggering 33% relative decrease in WER. It was observed that pre-training the DNN only provides a net reduction in WER of less than 1%. Hence, it was deemed to be not so critical for large data-sets. However, it may be far more useful for under-resourced languages.

The authors in [72] tried to use these models to solve the Google Voice Input speech recognition task. The model had four hidden layers with 2560 units per layer. The final softmax layer estimated the posteriors of 7969 HMM states. The input was 11 contiguous frames of 40 log filter-bank outputs with no temporal derivatives. Each hidden layer was pre-trained for one epoch as an RBM. The DNN system achieved 23% relative reduction in WER compared to the GMM systems.

In [73], the authors explore different optimization techniques to speed up the training of these huge DNN-based systems. Methods like parallelization of gradient computation and using low-rank matrix factorization to reduce the number of parameters are studied. They obtained a 3x improvement in speed with 33% reduced parameters.

The authors in [74] use dropout, a technique to prevent over fitting in neural networks along with rectified linear units (tend to over fit) easily on the LVCSR task. They obtained an improvement of 4.2% relative over the pre-trained DNN system. A careful analysis of this phenomenon was done in [75], where the authors noted an increase in sparsity and dispersion with the use of rectified linear unit-based hidden layers.

With the increased popularity of deep learning, denoising auto-encoders have also experienced resurgence. De-noising encoders are a simple learning technique, which learn to predict uncorrupted data from corrupt data using a feed-forward neural network with an optional bottleneck layer with a small dimension [76]. Experiments that reconstruct the clean speech from reverberant speech [77] have shown improvements and were also found to be effective for LVCSR systems. They have also been used to learn acoustic events in [78].

Very recent works in [79,80] uses bidirectional recurrent DNNs to build a first-pass LVCSR system. HMM-free approach to train system that directly predicts the transcription text from the audio utterance has been demonstrated in [79]. The work uses long short term memory (LSTM) neural network architecture along with the connectionist temporal classification (CTC) objective. Initial experiments on the Wall Street Journal corpus provide state-of-the-art accuracy. In [81], the authors extended the work stated in [79] by proposing

a modified prefix-search decoding algorithm. Experimental results were promising though the state-of-the-art performance was not obtained.

5. CONCLUSION

In this review article, a brief introduction about speech recognition and ML techniques are given to provide an insight to a novice reader in this domain. Both ASR and ML streams of research have been complementing each other since recent past, as both these paradigms are deeply ingrained within each other. ML in ASR was initiated with ANN-based speech recognizer and followed by numerous hybrid HMM/ANN systems. However, the momentum was lost due to the difficulty in learning techniques adopted. This difficulty has recently been overcome with the recent advancements in deep learning. HMM/GMM systems are being slowly overtaken as the most widely used learning models for ASR by DNN/HMM systems, which are shown to have obtained significant performance gains. Designing efficient deep learning architectures and algorithms that are scalable and robust with uncertain and incomplete data is a real challenge. Very recent work on CTC-based LVCSR gives an exciting path forward for LVCSR, doing away with the complexity of HMM-based infrastructures.

ACKNOWLEDGEMENTS

We are grateful to the anonymous reviewers for their valuable suggestions.

REFERENCES

1 H. Dudley, "The vocoder," *Bell Labs Rec.*, Vol. 17, pp. 122–6, 1939.

2 H. Dudley, R. R. Ryes, and S. A. Watkins, "A synthetic speaker," *J. Franklin Inst.*, Vol. 227, pp. 739–64, 1939.

3 K. H. Davis, R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits," *J Acoust Soc Amer.*, vol. 24, no. 6, pp. 637–42, Nov. 1952.

4 H. F. Olson, and H. Belar, "Phonetic typewriter," *J Acoust Soc Amer.*, vol. 28, no. 6, pp. 1072–81, Nov. 1956.

5 D. B. Fry, "Theoretical aspects of the mechanical speech recognition," *J. Br. Inst. Radio Eng.*, Vol. 19, no. 4, pp. 211–29, 1959.

6 T. K. Vintsyuk, "Speech discrimination by dynamic programming," *Kibernetika*, Vol. 4, pp. 81–8, 1968.

7 L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker independent recognition of isolated words using clustering techniques," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. 27, pp. 336–49, Aug. 1979.

8 J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. 38, pp. 1870–8, Nov. 1990.

9 S. K. Sahoo, T. Choubisa, and S. R. M. Prasanna, "Multimodal biometric person authentication: a review," *IETE Tech. Rev.*, Vol. 29, no. 1, pp. 54–75, May, 2012.

10 D. Pati, and S. R. M. Prasanna, "Speaker recognition from excitation source perspective," *IETE Tech. Rev.*, Vol. 27, no. 2, pp. 138–57, Sep. 2010.

11 H. S. Jayanna, and S. R. M. Prasanna, "Analysis, feature extraction, modeling and testing techniques for speaker recognition," *IETE Tech. Rev.*, Vol. 26, no. 3, pp. 181–90, Sep. 2009.

12 Sachin Singh, Manoj Tripathy, and R. S. Anand, "Subjective and objective analysis of speech enhancement algorithms for single channel speech patterns of Indian and English languages," *IETE Tech. Rev.*, Vol. 31, no. 1, pp. 34–46, May, 2014.

13 Tom M. Mitchell, *Machine Learning*, New York, NY: McGraw Hill, International Edition, 1997.

14 J. Baker, "Stochastic modeling for automatic speech recognition," in *Speech Recognition*. R. Reddy, Ed. New York, NY: Academic Press, 1976, pp. 297–307.

15 F. Jelinek, "Continuous speech recognition by statistical methods," Proc. IEEE, Vol. 64, no. 4, pp. 532–57, 1976.

16 Baker, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, and D. OShgughnessy, "Research developments and directions in speech recognition and understanding. Part I," *IEEE Signal Process. Mag.*, Vol. 26, no. 3, pp. 75–80, May, 2009.

17 L. Rabiner, and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

18 B.-H. Juang, S. E. Levinson, and M. M. Sondhi, "Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains," *IEEE Trans. Inf. Theory*, Vol. 32, no. 2, pp. 307–9, Mar. 1986.

19 L. Deng, P. Kenny, M. Lennig, V. Gupta, F. Seitz, and P. Mermelsten, "Phonemic hidden Markov models with continuous mixture output densities for large vocabulary word recognition," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. 39, no. 7, pp. 1677–81, Jul. 1991.

20 Bilmes, "What HMMs can do," *IEICE Trans. Inf. Syst.*, Vol. E89-D, no. 3, pp. 869–91, Mar. 2006.

21 H. Bourlard, and C. J. Wellekens, "Links between Markov models and multilayer perceptrons," in *Advances in Neural Information Processing*, D.S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 502–10.

22 N. Morgan, and H. Bourlard, "Continuous speech recognition using multilayer perceptrons with hidden Markov models," in *Proceedings of the IEEE International Conference ASSP*, Albuquerque, NM, 1990, pp. 413–6.

23 Morgan, H. Hermansky, H. Bourlard, P. Kohn, and C. Wooters, "Continuous speech recognition using PLP analysis with multilayer perceptrons," in *Proceedings of the IEEE International Conference ASSP*, Toronto, ON, 1991, pp. 49–52.

24 A. Ganapathiraju, J. Hamaker, and J. Picone, "Hybrid SVM/HMM architectures for speech recognition", in *Proceedings of the Neural Information Processing Systems*, Denver, CO, 2000, pp. 504–7.

25 J. Stadermann, and G. Rigoll, "A hybrid SVM/HMM acoustic modeling approach to automatic speech recognition," in *Proceedings of the Interspeech*, Jeju island, Korea, 2004, pp. 661–4.

26 S. Zhang, A. Ragni, and M. Gales, "Structured log linear models for noise robust speech recognition," *IEEE Signal Process. Lett.*, Vol. 17, pp. 945–8, Nov. 2010.

27 T. K. Landauer, C. A. Kamm, and S. Singhal, "Teaching a minimally structured back propagation network to recognize speech," in *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, Seattle, Washington, 1987, pp. 531–6.

28 S. M. Peeling, and R. K. Moore, "Isolated digit recognition experiments using the multilayer perceptron," *Speech Commun.*, Vol. 7, pp. 403–9, Dec. 1988.

29 Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. 37, pp. 328–39, Mar. 1989.

30 R. L. Watrous, and L. Shastri, "Learning phonetic features using connectionist networks: an experiment in speech recognition," in *Proceedings of the First International Conference on Neural Networks*, San Diego, California, 1987, pp. 381–8.

31 L. Elman, "Finding structure in time," CRL Tech., UCSD, Tech. Rep., La Jolla, California, 1988.

32 M. L. Jordan, "Serial order: A parallel distributed processing approach," UCSD, Tech. Rep. 8604, La Jolla, California, 1986.

33 T. Robinson, and F. Fallside, "A recurrent error propagation network speech recognition system," *Comput. Speech. Lang.*, pp. 259–74, May, 1991.

34 M. Richard, and R. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Comput.*, Vol. 3, pp. 461–83, Dec. 1991.

35 S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco. "Connectionist probability estimators in HMM speech recognition," *IEEE Trans. Speech Audio Process.*, Vol. 2, no. 1, pp. 161–74, Jan. 1994.

36 M. J. D. Powell, "Radial basis functions for multivariable interpolation: a review," *Algorithms for Approximation: in Proceedings of the IMA*, Shrivenham, England, 1987, pp. 143–67.

37 E. Singer, and R.P. Lippmann, "A speech recognizer using radial basis function neural networks in an HMM framework," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Montreal, Que., Canada, Vol. 1, 1992, pp. 629–32.

38 J. Hennebert, C. Ris, H. Bourlard, S. Renals, and N. Morgan, "Estimation of global posteriors and forward-backward training of hybrid HMM/ANN systems," in *Proceedings of the Eurospeech*, Rhodes, Greece, Vol. 4, 1997, pp. 1951–4.

39 H. Franco, M. Cohen, N. Moran, D. Rumelhart, and V. Abrash, "Context- dependent connectionist probability estimation in a hybrid hidden Markov model-neural net speech recognition system," *Comput. Speech Lang.*, Vol. 8, pp. 211–22, Jul. 1994.

40 H. Franco, and V. Digalakis, "Temporal correlation modeling in a hybrid neural network/hidden Markov model speech recognizer," in *Proceedings of the Eurospeech*, Madrid, Spain, 1995, pp. 1681–4.

41 M. M. Hochberg, S. J. Renals, A. J. Robinson, and G. D. Cook, "Recent improvements to the ABBOT LVCSR system," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Detroit, 1995, pp. 69–72.

42 D. Yu, T. Huang, and D. W. Chen, "A multi-stage NN/HMM hybrid method for high performance speech recognition," in *Proceedings of the ICSLP*, Yokohama, Japan, 1994, pp. 1503–6.

43 W. Y. Chen, S. H. Chen, and C. J. Lin, "A speech recognition method based on the sequential multi-layer perceptrons," *Neural Netw.*, Vol. 9, no. 4, pp. 655–69, Jun. 1996.

44 Y. Yan, M. Fanty, and R. Cole, "Speech recognition using neural networks with forward-backward probability generated targets," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Munich. Germany, 1997, pp. 3241–4.

45 F. Grezl, M. Karafiat, K. Stanislav, and J. Cernocky. "Probabilistic and Bottle-neck Features for LVCSR of Meetings," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Honolulu, HI, 2007, pp. 757–60.

46 A. Stolcke, F. Grezl, M-Y Hwang, X. Lei, N. Morgan, and D. Vergyri. "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Toulouse, 2006, pp. 321–4.

47 L. Toth, J. Frankel, G. Gosztolya, and S. King. "Cross-lingual portability of MLP-based tandem features - a case study for English and Hungarian," in *Proceedings of the Interspeech*, Brisbane, Australia, 2008, pp. 2695–98.

48 D. Imseng, H. Bourlard, and M. Magimai.-Doss, "Towards mixed language speech recognition systems," in *Proceedings of the Interspeech*, Makuhari, Japan, 2010, pp. 278–81.

49 N. Thang Vu, F. Metze, and T. Schultz, "Multilingual bottle-neck features and its applications for under-resourced languages," In *Proceedings of the SLTU*, Cape Town, South Africa, 2012, pp. 90–3.

50 M. J. J. Premkumar, N. Thang Vu, and T. Schultz. "Experiments towards a better LVCSR system for Tamil," in *Proceedings of the Interspeech*, Lyon, France, 2013, pp.1012–16.

51 M. D. Yu, and L. Deng, "Investigation of full-sequence training of deep belief networks for speech recognition," in *Proceedings of the Interspeech*, Makuhari, Japan, 2010, pp. 2846–49.

52 C. Cortex, and V. Vapnik. "Support vector networks," *Mach. Learn.*, Vol. 20, no. 3, pp. 273–97, Sep. 1995.

53 A. Ganapathiraju, J. E. Hamaker, and J. Picone, "Applications of support vector machines to speech recognition," *IEEE Trans Signal Process.*, Vol. 52, pp. 2348–55, Aug. 2004.

54 C. Sekhar, W. F. Lee, K. Takeda, and F. Itakura, "Acoustic modelling of subword units using support vector machines," in *Proceedings of the Workshop on Spoken Language Processing*, Mumbai, India, 2003, pp. 79–86.

55 J. Padrell-Sendra, D. Martin-Iglesias, and F. Diaz-de-Maria, "Support vector machines for continuous speech recognition," In *Proceedings of the 14th European Signal Processing Conference*, Florence, Italy, 2006, pp. 434–39.

56 S. J. Young, N. H. Russell, and J. H. S. Thornton. "Token Passing: a conceptual model for Connected Speech Recognition Systems," CUED Cambridge University, UK, Tech. Rep., 1989.

57 J. C. Platt. *Advances in Kernel Methods: Support Vector Learning, chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization*, Cambridge: MIT Press, 1999, pp. 185–208.

58 Ech-Cherif, M. Kohili, A. Benyettou, and M. Benyettou, "Lagrangian support vector machines for phoneme classification," In *Proceedings of the 9th International Conference on Neural Information Processing*, Orchid country club, Singapore, Vol. 5, 2002, pp. 2507–36.

59 S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modeling," in *Proceedings of the workshop on Human Language Technology*, Plainsboro, NJ, 1994, pp. 307–12.

60 L. E. Baum, and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Stat.*, Vol. 37, no. 6, pp. 1554–63, Dec. 1966.

61 P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. Ser. B*, Vol. 39, no. 1, pp. 1–38, 1977.

62 M. Gales, and S. Young, "Robust continuous speech recognition using parallel model combination," *IEEE Trans. Speech Audio Process.*, Vol. 4, no. 5, pp. 352–9, Sep. 1996.

63 D. Povey, L. Burget et al.,"Subspace Gaussian mixture models for speech recognition," in *Proceedings of the ICASSP*, Dallas, TX, 2010, pp. 4330–33.

64 G. Hinton, and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, Vol. 313, no. 5786, pp. 504–7, Jul. 2006.

65 G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, Vol. 18, pp. 1527–54, Jul. 2006.

66 Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, Vol. 2, no. 1, pp. 1–127, Nov. 2009.

67 N. Morgan, Q. Zhu, A. Stolcke, K. Sonmez, S. Sivadas, T. Shinozaki, M. Ostendorf, P. Jain, H. Hermansky, D. Ellis, G. Doddington, B. Chen, O. Cretin, H. Bourlard, and M. Athineos, "Pushing the envelope Aside [speech recognition]," *IEEE Signal Process. Mag.*, Vol. 22, no. 5, pp. 81–8, Jul. 2005.

68 L. Deng, D. Yu, and J. Platt, Scalable stacking and learning for building deep architectures, in *Proceedings of the IEEE International Conference. Acoust. Speech, Signal Process*, Kyoto, Japan, 2012, pp. 2133–6.

69 C.-H. Lee, "From knowledge-ignorant to knowledge-rich modeling: A new speech research paradigm for next-generation automatic speech recognition," in *Proceedings of the International Conference Spoken Lang, Process*, Jeju Island, Korea, 2004, pp. 109–11.

70 G. Dahl, D. Yu, L.Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio Speech Lang. Process.*, Vol. 20, no. 1, pp. 30–42, Jan. 2012.

71 F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proceedings of the Interspeech*, Florence, Italy, 2011, pp. 437–40.

72 N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "An application of pretrained deep neural networks to large vocabulary conversational speech recognition," Department of Computer Science, University of Toronto, Tech. Rep. 001, Toronto, Canada, 2012.

73 T. N. Sainath, B. Kingsbury, H. Soltau, and B Ramabhadran, "Optimization Techniques to improve training speed of deep neural networks for large speech tasks," *IEEE Trans. Audio Speech Lang. Process.*, Vol. 21, no. 11, pp. 2267–76, Nov. 2013.

74 G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proceedings of the ICASSP*, Vancouver, Canada, 2013, pp. 8609–13.

75 A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier non linearity improved neural network acoustic models," in *Proceedings of the ICML Workshop on Deep Learning for Audio, Speech, and Language Processing*, Atlanta, GA, 2013. Available: http://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf

76 P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "extracting and composing robust features with denoising auto encoder," in *Proceedings of the twenty-fifth International conference on Machine Learning*, Helsinki, Finland, 2008, pp. 1096–103.

77 T. Ishii, H. Komiyama, T. Shinozaki, Y. Horiuchi, and S. Kuroiwa, "Reverberant speech recognition based on denoising auto encoder," in *Proceedings of the Interspeech*, Lyon, France, 2013, pp. 3512–16.

78 N. Jaitley, and G. E. Hinton, "A new way learn acoustic event," *Adv. Neural Inf. Process. Syst.* 2011. vol. 24. Available: http://www.cs.toronto.edu/~hinton/absps/capsules_speech.pdf.

79 A. Graves, and N. Jaitly, "Towards End-to-End Speech Recognition with Recurrent Neural Networks," in *Proceedings of the ICML*, Beijing, China, 2014, pp. 1764–72.

80 Martin Wollmer, B. Schuller, and G. Rigoll, Probabilistic ASR feature extraction applying context-sensitive connectionist temporal classification networks, 2013. Available: http://www.mmk.ei.tum.de/publ/pdf/13/13woe1.pdf

81 Andrew L. Maas, Awni Y. Hannun, Daniel Jurafsky, and Y. Andrew, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs," Cornell University lib. ArXiv:1408, Aug. 2014.

Authors

Jayashree Padmanabhan received her BE (Hons) degree in Electronics and Communication from Madurai Kamaraj University, masters (Electronics Engineering) and PhD (Computer Science Engineering) degrees from Anna University, Chennai. She is currently an associate professor in the Department of Computer Technology, MIT Campus of Anna University, Chennai, India. She has rich teaching and research experience and she has nearly 20 reputed journal/conference publications. Her research interests include wired/wireless network security, cryptographic algorithms, machine learning, parallel computing, and E-learning.

E-mail: pjshree12@gmail.com

Melvin Jose Johnson Premkumar is currently pursuing his masters degree in Computer Science at Stanford University, specializing in Artificial Intelligence. He did his bachelor's degree in Computer Science and Engineering at Madras Institute of Technology, Anna University. His interests include speech recognition, machine learning, and artificial intelligence.

E-mail: jmelvinjose73@yahoo.com

DOI: 10.1080/02564602.2015.1010611; Copyright © 2015 by the IETE