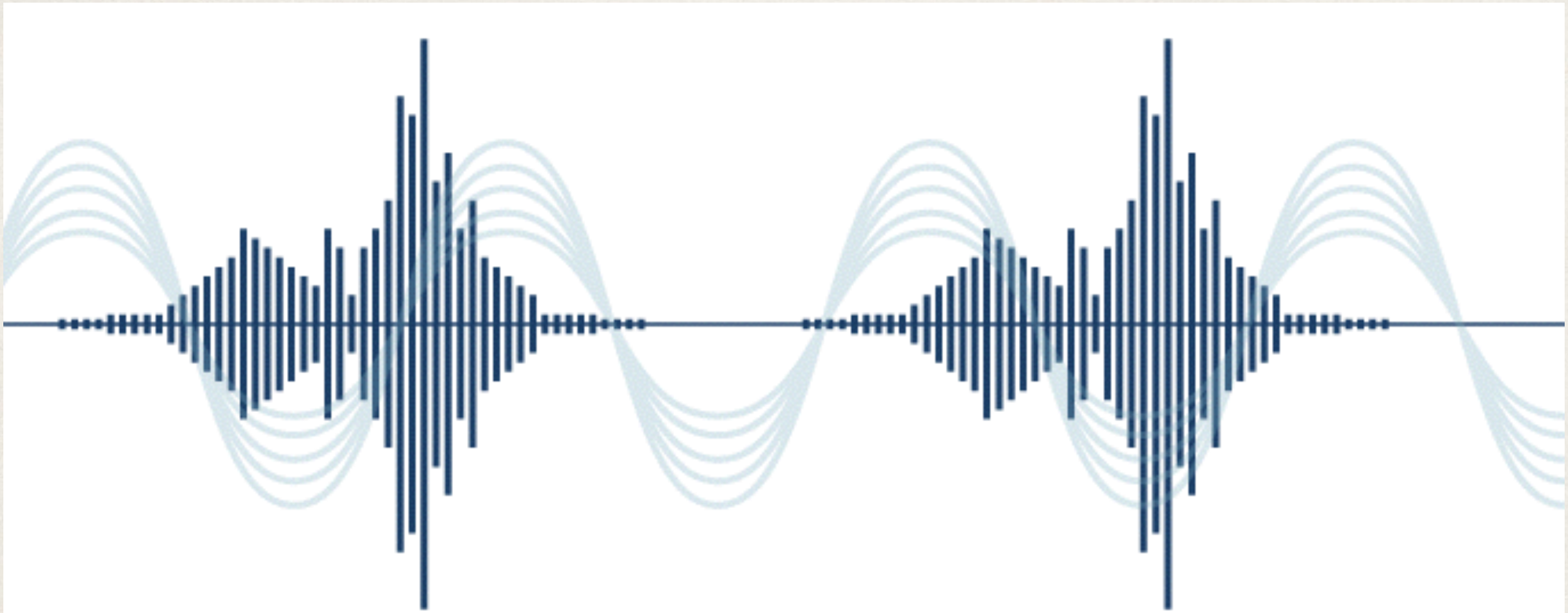


Hindi Automatic Speech Recognition on Distributed Systems

BTP Mid Term Presentation

Pratyush Sharma Kartik Parnami Jigya Yadav

"As far as the customer is concerned, the interface is the product."

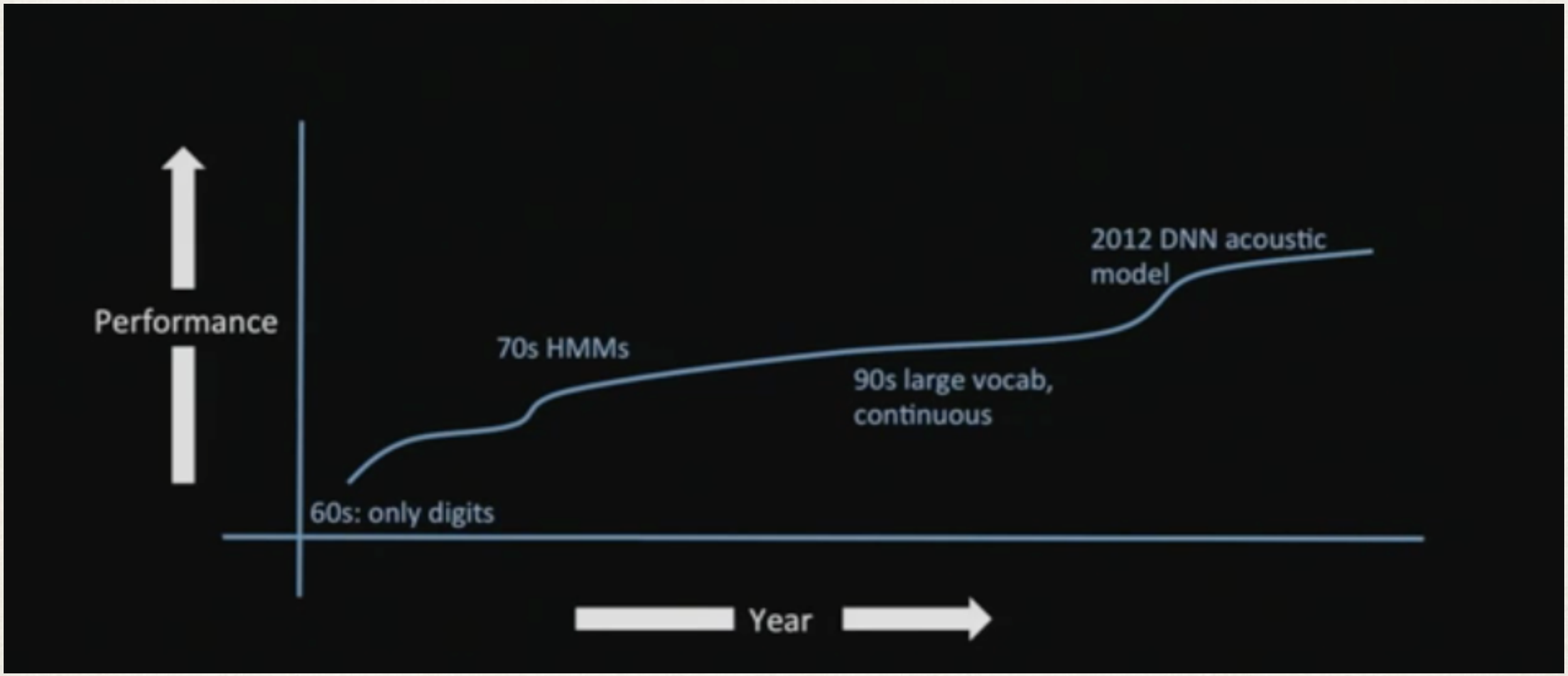
–Jeff Raskin

Problem Statement

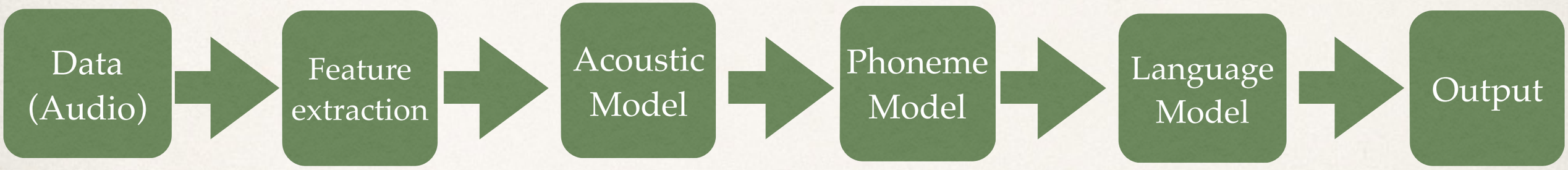
- The problem we will tackle is Human Speech Recognition for Hindi language using Deep Neural Networks.

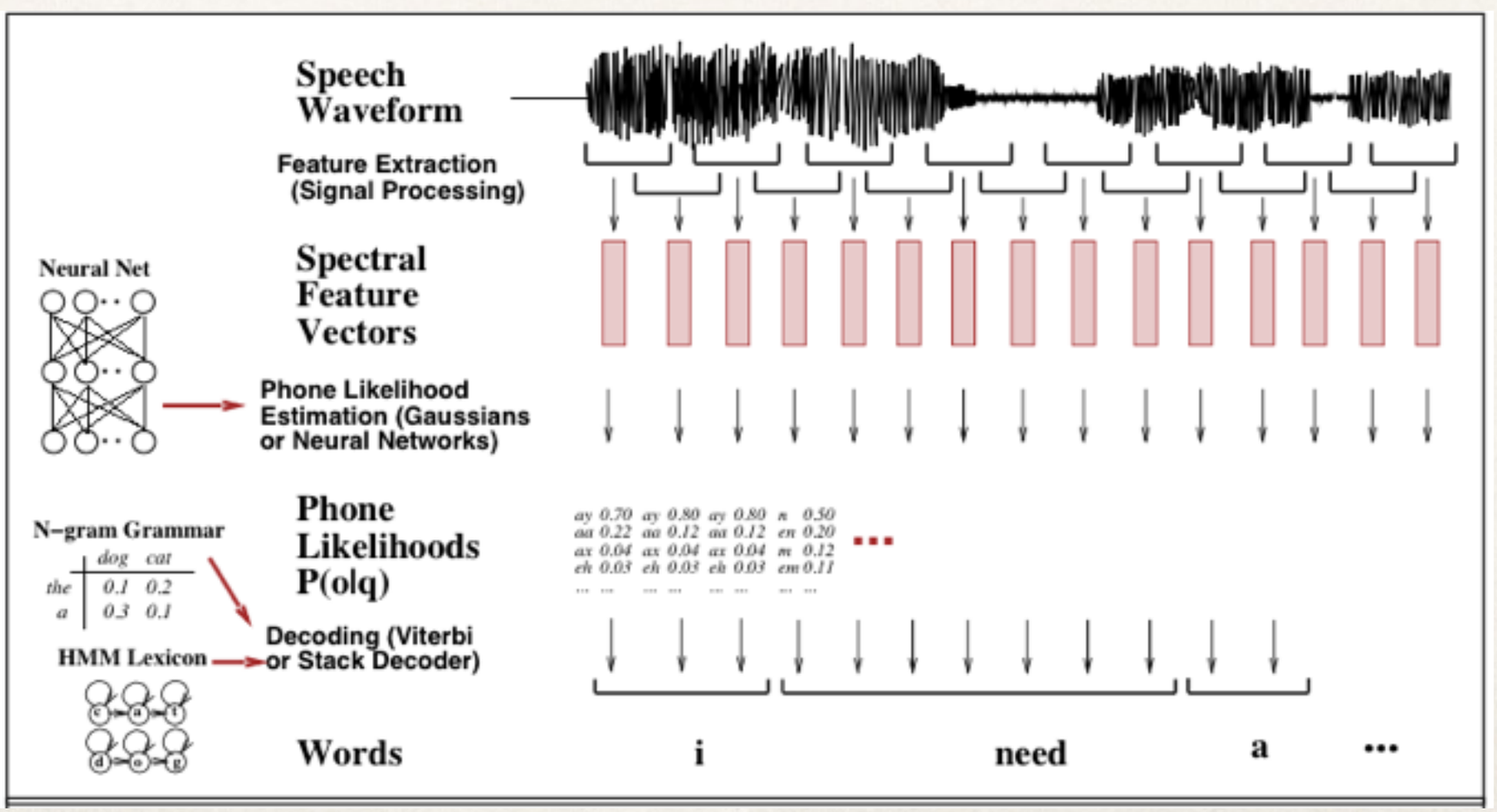
- We will also develop an interface for recording of speech which can segregate the recording into separate words.

History Of Speech Recognition



Automatic Speech Recognition Model





Feature Extraction

- Provides a compact representation of the speech waveform
- Should minimise the loss of information that discriminates between words
- Should provide a good match with the distributional assumptions made by the acoustic model
- Feature vector used in proposed system: MFCC (Mel Frequency Cepstral Coefficients)

MFCC Feature Extraction

MFCC Specifications

Mohammad et. al (2013) (LPC and MFCC Performance Evaluation with Artificial Neural Network for Spoken Language Identification) studied the error % for MFCC of different orders, leading to results proving that error % dipped when MFCC order is around the 10th order.

Mfcc Order	Training Error %	Testing Error %
3	20	20
5	15	15
7	10	10
10	0	0
15	5	5
20	4	4

MFCC Specifications (Contd.)

Zheng, Song et. al (2013) (Comparison of Different Implementations of MFCC) showed by research that the recognizer reaches the maximal performance at the filter number K = 35. Too few or too many filters do not result in better accuracy.

Number of MFCC Filters	Performance %
25	67.39
30	67.73
35	68.01
40	67.84
45	67.86

Acoustic Model

- Integrates knowledge about acoustics and phonetics
- Used to predict the phonemes that occur in input data
- Previously Gaussian Mixture Model(GMM) based Hidden Markov Models were used for acoustic modelling
- **Shortcoming of GMM-HMM model:** Statistically inefficient for modelling data that lie on or near a non-linear manifold in space
- Proposed system uses **Deep Neural Network- Hidden Markov Model(DNN-HMM)** hybrid architecture

Hidden Markov Models

- An HMM is a doubly stochastic process with an underlying stochastic process that is not observable, but can only be observed through another set of stochastic processes that produce the sequence of observed symbols.
- Specified by the following components:

$Q = q_1 q_2 ... q_N$	a set of N states
$A = a_{11} a_{12} ... a_{n1} ... a_{nn}$	a transition probability matrix
$O = o_1 o_2 ... o_T$	a sequence of T observations
$B = b_i(o_t)$	a sequence of observation likelihoods
q_0, q_F	a special start state and final state

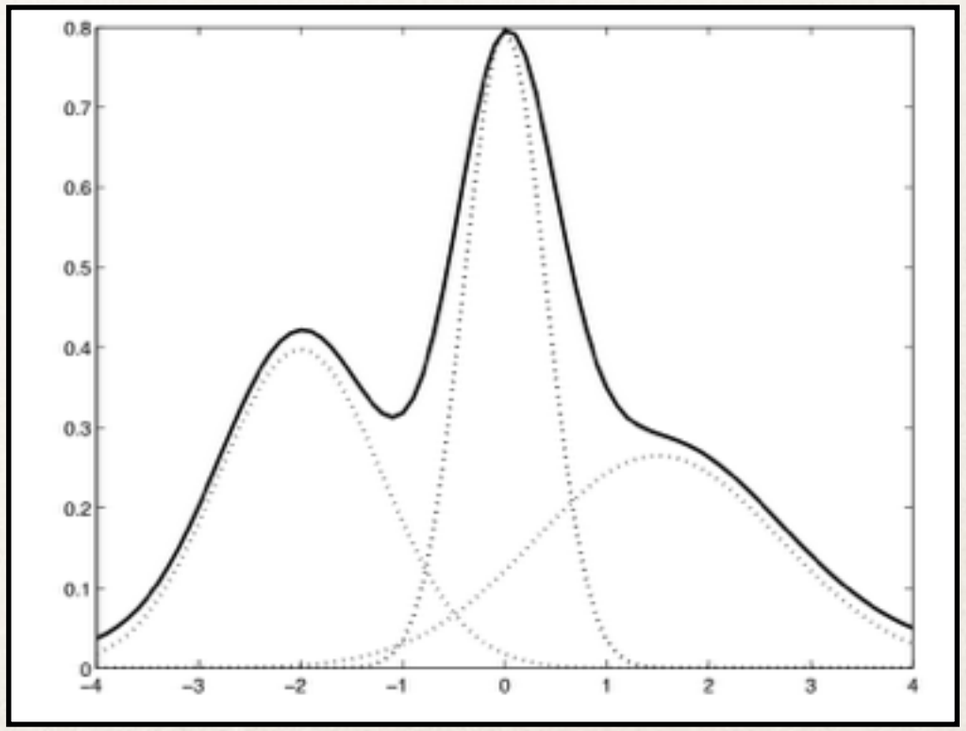
Hidden Markov Models (Cont.)

- 3 fundamental problems of HMMs:
  - Computing Likelihood: Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O \mid \lambda)$. Solved by using forward algorithm.
  - Decoding: Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q. Solved by using Viterbi algorithm.
  - Learning: Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B. Solved by using Baum-Welch algorithm.

Gaussian Mixture Models

- GMMs are used to represent frame-based speech features
- Used for estimating acoustic likelihoods i.e. the probability that HMM state *j* generates the value of a single dimension of a feature vector
- GMMs are a weighted sum of multivariate Gaussians

$$f(x|\mu,\Sigma) = \sum_{k=1}^{M} c_k \frac{1}{\sqrt{2\pi|\Sigma_k|}} exp[(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)]$$



Parameter Estimation of GMM

- The output likelihood function is defined as:

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} \frac{1}{\sqrt{2\pi|\Sigma_{jm}|}} exp[(x-\mu_{jm})^T \Sigma_{jm}^{-1}(o_t-\mu_{jm})]$$

- In order to estimate the mean and variance parameters of the GMM, we use expectation maximization algorithm.

$$c_m^{(j+1)} = \frac{1}{N}\sum_{t=1}^{N} h_m^{(j)}(t),$$

$$\boldsymbol{\mu}_m^{(j+1)} = \frac{\sum_{t=1}^{N} h_m^{(j)}(t) x^{(t)}}{\sum_{t=1}^{N} h_m^{(j)}(t)},$$

$$\boldsymbol{\Sigma}_m^{(j+1)} = \frac{\sum_{t=1}^{N} h_m^{(j)}(t)[\mathbf{x}^{(t)} - \boldsymbol{\mu}_m^{(j)}][x^{(t)} - \boldsymbol{\mu}_m^{(j)}]^T}{\sum_{t=1}^{N} h_m^{(j)}(t)},$$

M-Step

$$h_m^{(j)}(t) = \frac{c_m^{(j)} \mathcal{N}(\mathbf{x}^{(t)}; \boldsymbol{\mu}_m^{(j)}, \boldsymbol{\Sigma}_m^{(j)})}{\sum_{i=1}^{n} c_i^{(j)} \mathcal{N}(\mathbf{x}^{(t)}; \boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)})}.$$

E-Step

GMM-HMM models in speech recognition

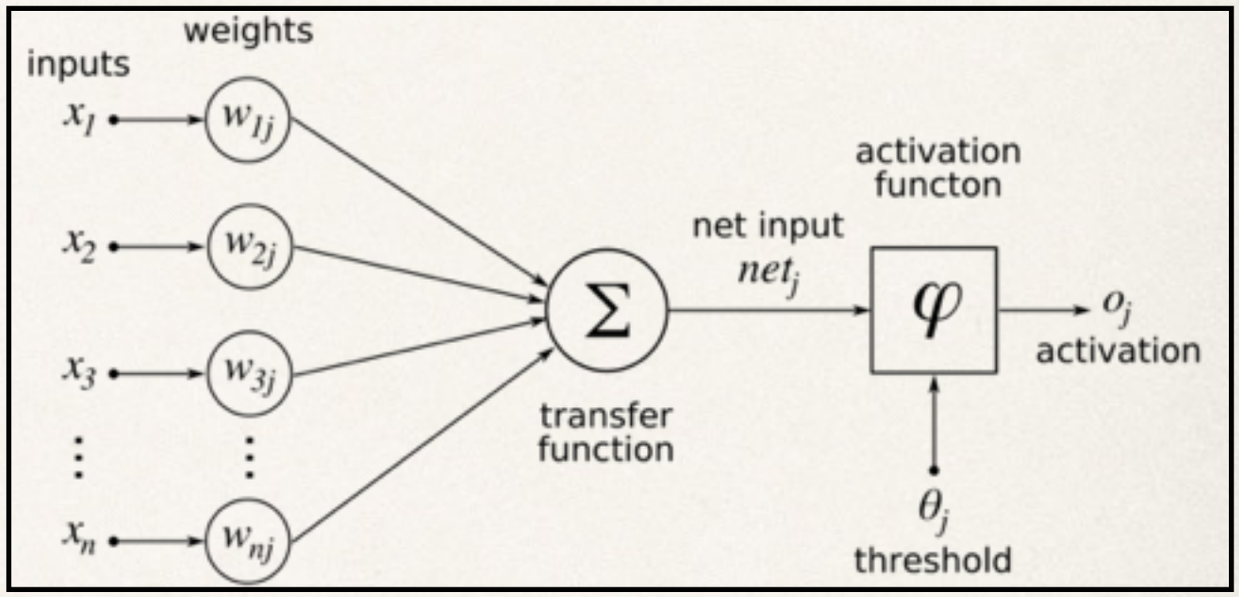
- GMM-HMM is a statistical model that describes two dependent random processes, an observable process and a hidden Markov process
- Observation sequence is generated by state according to Gaussian Mixture distribution- Used to compute acoustic likelihood $b_j(o_t)$ of frame
- HMM used for decoding: "Given a string of acoustic observations, how should we choose the string of words which has the highest posterior probability?"
- Key Concept:

$$\hat{W} = \underset{W \in \mathcal{L}}{\operatorname{argmax}} \overbrace{P(O|W)}^{\text{likelihood}} \overbrace{P(W)}^{\text{prior}}$$

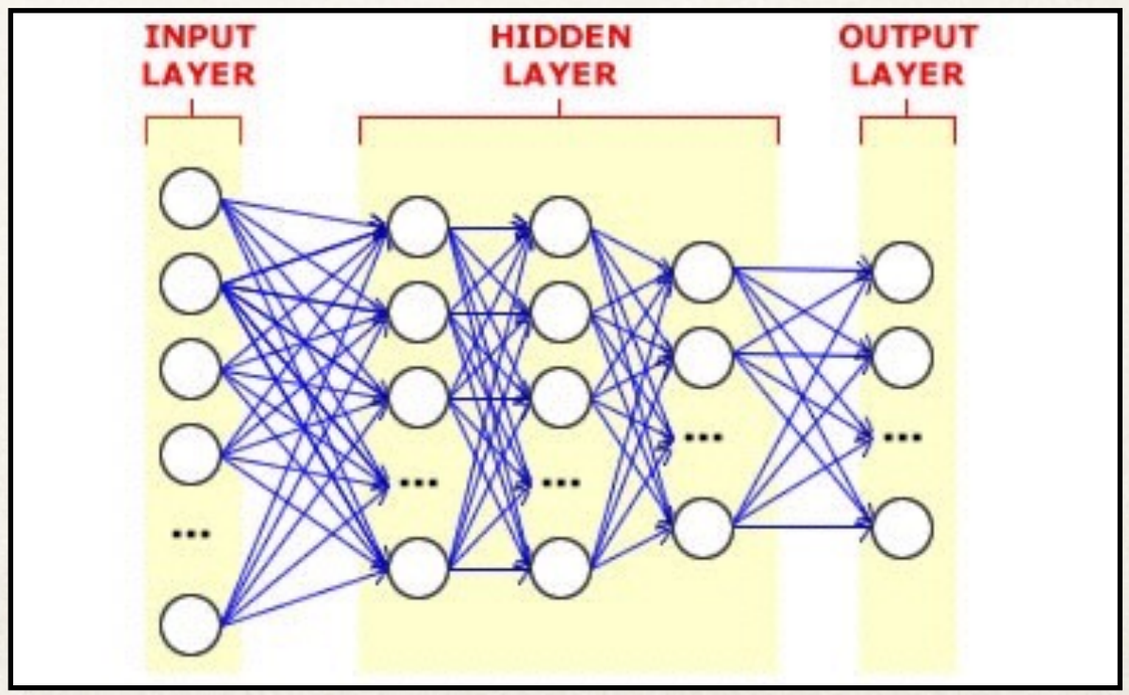
- Use forward algorithm and Viterbi algorithm to compute P(O | W) and language models to compute P(W)

Deep Learning

- Model of a "Neuron":



- Deep neural network is a feed-forward artificial neural network with multiple hidden units between input and output

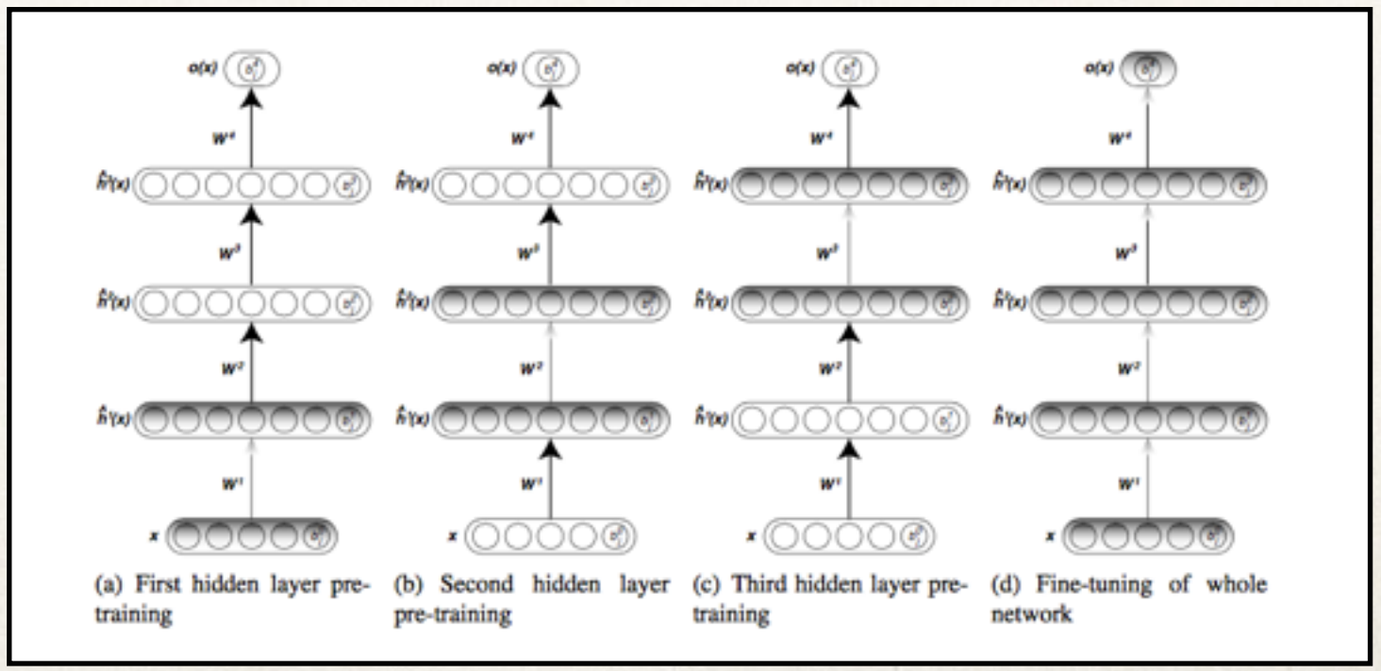


Why Deep Learning?

- DNNs have ability to learn complicated feature representations and classifiers jointly
- Learn much better models of data that lie on or near a non-linear manifold
- Performance does not saturate with increase in training data
- Deep learning proven to be better than GMM-HMM: e.g..In Large Vocabulary Continuous Speech Recognition with Context Dependent DBN-HMMS by George E. Dahl et al., an improvement of absolute sentence accuracy improvements of 9.2% over GMM-HMMs was obtained.

Training of Deep Neural Networks

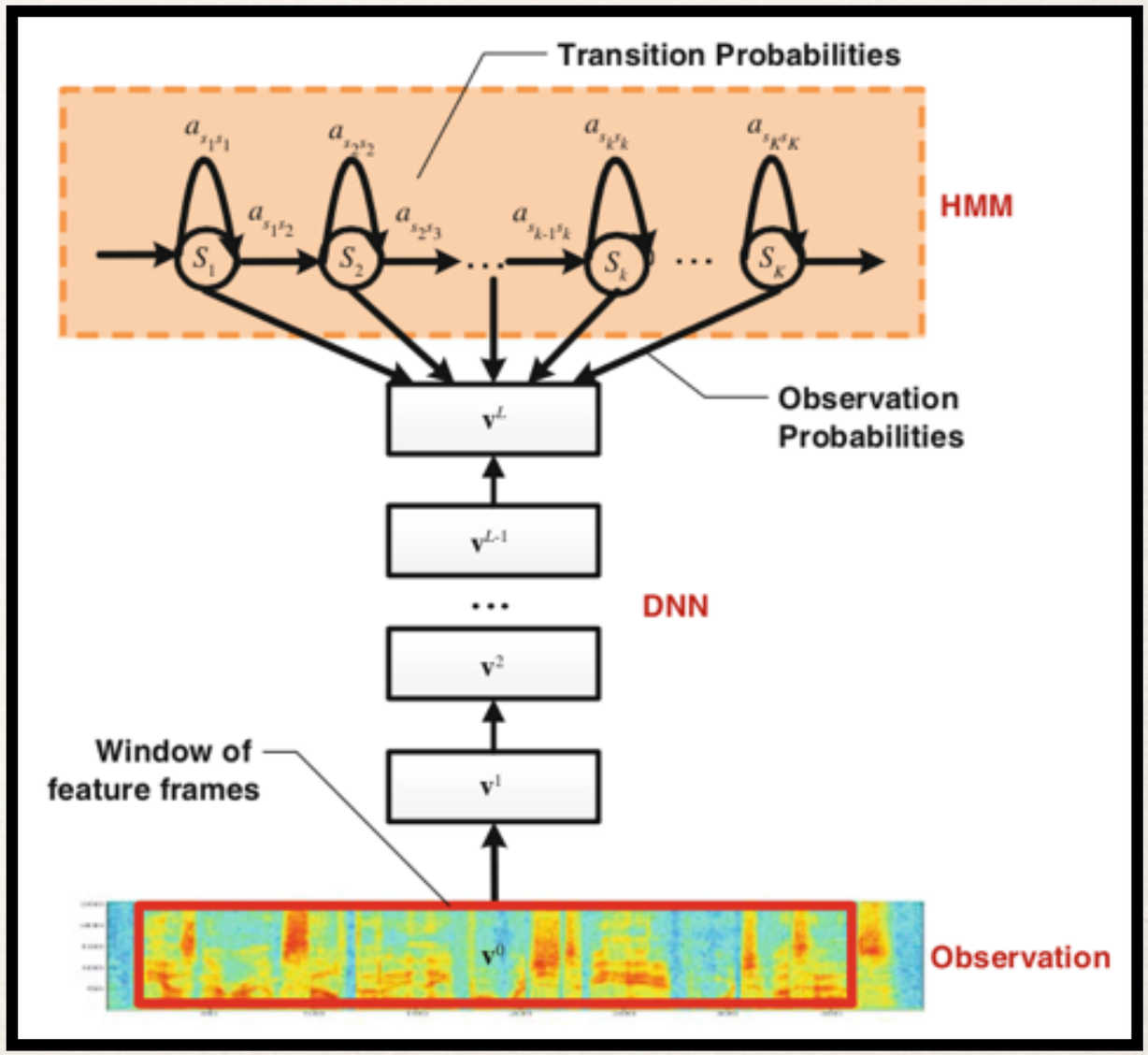
- Greedy layer wise training:
  - Pre-training one layer at a time in a greedy way using unsupervised learning. Until a stopping criteria is met, iterate through training database by:
    - mapping input training sample x_t to representation $h_{i-1}(x_t)$ and hidden representation $hi(x_t)$
    - updating parameters b_{i-1}, b_i, and W_i of layer i
- Fine-tuning the whole network with supervised learning, back-propagation and gradient descent



(a) First hidden layer pre-training (b) Second hidden layer pre-training (c) Third hidden layer pre-training (d) Fine-tuning of whole network

DNN-HMM Hybrid Models

- Dynamics of the speech signal modelled with HMMs
- Observation probabilities estimated through DNNs



Estimation of output using DNN-HMM

- Decoded word sequence *w is* determined as:

$$\hat{w} = \arg\max_w p(w|\mathbf{x}) = \arg\max_w p(\mathbf{x}|w)\,p(w)/p(\mathbf{x})$$
$$= \arg\max_w p(\mathbf{x}|w)\,p(w),$$

where $p(w)$ is the language model (LM) probability, and

$$p(\mathbf{x}|w) = \sum_q p(\mathbf{x}|q, w)\,p(q|w)$$
$$\approx \max\pi(q_0) \prod_{t=1}^{T} a_{q_{t-1}q_t} \prod_{t=0}^{T} p(q_t|\mathbf{x}_t)/p(q_t)$$

where $p(x \mid w)$ is the acoustic model probability, $p(q_t \mid x_t)$ is computed from the DNN, $p(q_t)$ is the state prior probability.

$\pi(q_0)$ and $a_{q_{t-1}q_t}$ are the initial state probability and the state transition probability, which are determined by the HMM.

DNN proposed structure

- Our proposed DNN will contain 3-4 hidden layers, with 13-39 input neurons depending upon the number of MFCCs, a varying number of hidden neurons depending upon whatever number best fits the design and output neurons equal in number to the number of phonemes to be recognized.

- We will also compare the DNN's performance with an RNN which has been proved to perform better than DNN in some cases.

Training of DNN-HMM Models

- Train GMM-HMM model
- Align GMM-HMM senones with DNN output data
- We can then generate the featureSenoneIDPairs from the alignment and use them to train the DNN.
- GMM-HMM model hmm0 we can also generate an HMM hmm, A simple approach is to replace each GMM (which models one senone) in hmm0 with a (pseudo) single one-dimensional Gaussian.The value of each senone's mean is set to the corresponding senoneID. Using this trick, evaluating each senone is equivalent to a table lookup of the features (log-likelihood) produced by the DNN with the index indicated by the senoneID
- GMM free approach involves segmenting the utterances (called flat-start) and using that information as the training label.

Phoneme Model

- A phonetic dictionary contains a mapping from words to phones. This mapping is not very effective. For example, only two to three pronunciation variants are noted in it, but it's practical enough most of the time.

- The dictionary is not the only variant of mapper from words to phones. It could be done with some complex function learned with a machine learning algorithm.

- Dictionary generated from the data in hand

Language Model

- Speech recognisers seek the word sequence W which is most likely to be produced from acoustic evidence A

$$P(\hat{W}|A) = \max_{W} P(W|A) \propto \max_{W} P(A|W)P(W)$$

- Language Models assign a probability estimate to word sequences $W = \{w_1, w_2, \ldots, w_n\}$
- Language model used in proposed system: n-gram models
- n-gram models use the previous (n-1) words to estimate the probability of each word
- n-gram parameters are estimated by counting n-tuples in text corpora

Training Data - Testing and Validation

- Testing and Validation Using Buckeye Corpus
- The Buckeye Corpus of conversational speech contains high-quality recordings from 40 speakers in Columbus OH conversing freely with an interviewer. The speech has been orthographically transcribed and phonetically labeled.
- Hindi data from past research done from TIFR and NIT kurukshetra

Synthesized data

Voice

Add Noise

Synthesized Data

Data Generation module

- Data generation for hindi training
- Developing single speaker training data using smart module
- Training data is phonetically transcribed
- Module will give speaker line to read and do word by word timing and phoneme phoneme timing

Theano

- A CPU and GPU Math Compiler in Python
- Python library that allows definition, optimisation and evaluation of mathematical expressions and multi-dimensional arrays
- Combines aspects of Computer Algebra System(CAS) with aspects of optimising compiler for faster evaluation of complicated mathematical expressions
- Includes CUDA code generator for computations using GPU
- Theano used to apply the speech recognition system to GPUs and distributed systems

Division Of Work

Jigya	Kartik	Pratyush
Feature Extraction & DNN Architecture	DNN - Implementation and Testing	HMM Model for Phonemes & training of GMM-HMM Model for alignments

Present Work and Future Plan

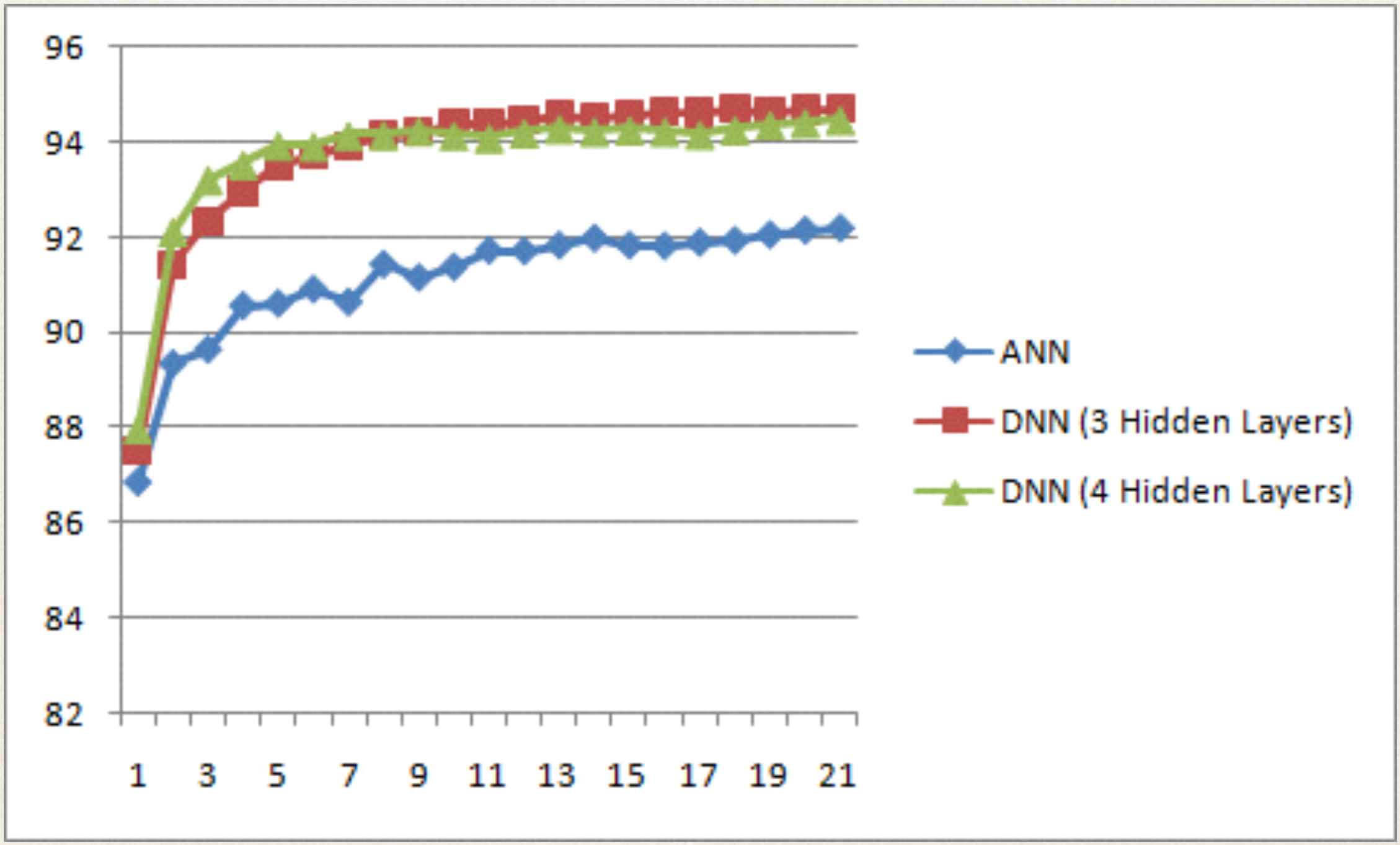
Present Work

- Implemented and Tested individual components - Deep neural network, GMM-HMM model, MFCC

Future Plan

- Implementation of DNN-HMM hybrid architecture
- Training the network with English and then Hindi

Results of DNN training on MNIST handwritten digit recognition data



Developments in Hindi Voice Recognition

- CDAC ASR for farmers etc.

- Samudravijaya, SS Agrawal, "Hindi Speech Database", Proc. Int. Conf. Spoken language processing, ICSLP00, October Beijing 2000, CDROM:00192.pdf

- N. Shrotriya, R. Verma, S.K. Gupta, S.S. Agrawal, "Durational characteristics of Hindi consonant clusters," Proc. ICSLP, vol. 4, 2427-2430, 1996

- G.V. Rao, J. Srichland, "Word boundary detection using pitch variations", Proc. ICSLP, Vol. 2, 813-816, 1996.

References

- Andrew Senior, Georg Heigold, Michiel Bacchiani, Hank Liao, GMM-FREE DNN TRAINING, 2012

- Li Deng, Dong Yu. "Automatic Speech Recognition", Springer, 2015

- Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic model- ing in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. 29(6), 82–97 (2012)

- Samudravijaya, SS Agrawal, “Hindi Speech Database”, Proc. Int. Conf. Spoken language processing, ICSLP00, October Beijing 2000, CDROM:00192.pdf

- A. Graves, A. Mohamed, G. Hinton. Speech Recognition with Deep Recurrent Neural Networks. ICASSP 2013, Vancouver, Canada

Thank you!!