Hybrid Deep Neural Network Hidden Markov Models (DNN-HMM) Systems for Isolated Word Recognition

Rendani Mbuvha
KTH, Royal Institute of Technology
rendani@kth.se

Nino Prekratić
KTH, Royal Institute of Technology
ninop@kth.se

Abstract

In recent years Automatic Speech Recognition (ASR) has gained increasing attention with the rise of voice controlled applications in mobile devices and even automobiles. Traditionally, ASR systems were built using Hidden Markov Models (HMMs) with Gaussian Mixture Model emissions. However, with the recent resurgence of Deep Neural Networks (DNNs), there has been an increase in their application in almost all Machine Learning domains, including ASR. This project assesses the accuracy of different DNN architectures in phoneme recognition and with an aim to build a Hybrid DNN-HMM System for isolated word recognition on the TIDIGITS dataset.

1 Introduction

Neural networks have emerged as a promising acoustic modelling approach in ASR in the late 1980s. Since than have been used for different tasks in speech recognition, such as phoneme classification, isolated word recognition and speaker adaptation. Compared to HMMs, they dont make any assumptions about statistical properties of features and ensure a discriminative training in an efficient way. Neural networks are very good in classifying short-time units such as individual phonemes and isolated words, but because of their lack of ability to model temporal dependencies, they are not very successful in continuous speech recognition tasks and even for slightly longer time units like an isolated word. In order to improve this aspect, different types of neural networks have been used for the speech recognition task, such as Recurrent Neural Networks (RNNs), Time Delay Neural Networks (TDNNs) and Deep Neural Network-Hidden Markov Model (DNN-HMM) Hybrid Systems [1].

DNN-HMM hybrid systems take advantages of both systems; they use DNNs strong representation learning power and HMMs sequential modelling ability, which as a result produces a system that outperforms conventional Gaussian mixture model (GMM) HMM systems in the tasks of both continuous and isolated speech recognition with a large vocabulary [3].

In this project, we are going to compare DNN networks with different architectures and features sets and further develop a DNN-HMM hybrid system for isolated word recognition.

2 Related Work

There is a lot of previous work done investigating and comparing performances of different speech recognition systems. Most of them deal with comparison of more conventional HMM model with some type of DNN model. In our work, we aim at comparing the performance between few different DNN network models and construct a DNN-HMM hybrid system.

In [4], authors present context dependent (CD) model for large-vocabulary speech recognition (LVSR) that uses pre-trained DNN-HMM hybrid architecture. They have shown that their system outperformed the traditional GMM-HMM system, which was trained using minimum phone error rate (MPE) and maximum-likelihood criteria (ML) with and absolute sentence accuracy improvement of 5% and 9.2% respectively. In [5], authors have performed speech-to-text transcription using their CD-DNN-HMM system. They have applied this system to a singlepass speaker-independent recognition on RT03 Fisher portion of phone-call transcription benchmark (Switchboard). Compared to CD-GMM-HMM system, the word error rate is reduced from 27.4% to 18.5%. In both of these cases, it is noted that even though the increase in accuracy compared to CD-GMM-HMM is large, the training process is comparatively very expensive and slow.

Authors of paper [6] have used two types (back propagation - BP and associative memory - AM) of Deep Belief Network (DBN) for acoustic modelling and they have investigated the influence of model depth and hidden layer size. Both architectures had over fitting methods implemented (bottleneck for BP-DBN and discriminative training for AM-DBN) and have outperformed other reported results on TIMIT core test set achieving a phone error rate (PER) of 23%.

Finally in [2], Hinton et al. are providing examples where few major research groups managed to significantly improve the performances of different ASR systems by substituting GMMs with DNNs. Authors opinion is that there is room for significant improvements in the future, because they think that there is no evidence by now that they are using most optimal type of hidden units and network architectures. Also, they hope to improve pre-training and fine-tuning algorithms in order to reduce the amount of computation and over fitting.

3 Method

As we focus on constructing a Hybrid DNN-HMM system for isolated word recognition we proceed by describing the conceptual frameworks behind Deep Neural Networks (DNNs) and Hidden Markov models(HMMs).

3.1 Artificial Neural Networks

An Artificial Neural Network, more specifically a Multilayer Perception (MLP), by definition is a directed graph that maps an input to an output by propagating the input through a series of non-linear transformations at the nodes of various layers of the network. The network attempts to learn an approximation of the output by minimizing a loss function which is often a function of the difference between the network's predicted outputs and the ground truth in a given set of training examples. This loss function typically also includes a regularization element, which attempts to prevent the network from over fitting the training examples. In recent years more advanced techniques for regularisation like dropout have also been used. Figure 1 shows a typical MLP.

3.1.1 Deep Neural Networks (DNNs)

DNNs are generally defined as neural networks with more than one hidden layer. The universal approximation theorem for neural networks [7] shows that an MLP with one hidden layer is sufficient to approximate any non-linear function provided it has enough nodes in the hidden layer. However when networks layer becomes wider, they become harder to parallelize and therefore harder to train. Also, as a consequence of speech, images and text data having an implied hierarchical representation of features, training results turn out to be better if network has learnt using multiple consecutive layers, rather than a single wide layer.

Recent developments in machine learning algorithms, computer software and computer hardware have led to efficient models for the training of neural networks with many layers of non-linear hidden

units and a big output layer. Deep neural networks are trained using the same backpropagation algorithm as single hidden layer MLPs ideally with an even greater emphasis on regularisation.

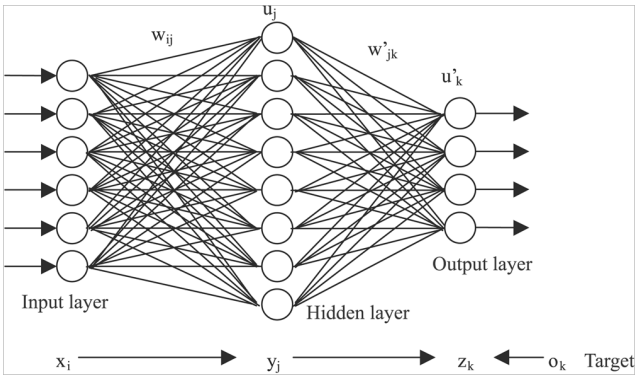

Figure 1: Artificial Neural Network

3.2 Hidden Markov Models (HMMs)

Hidden Markov Models (HMM) are simplistic Bayesian Networks that consist of two statistical processes,a Markov process for state transition and an emission process for generating observations. In an HMM the state of the Markovian process is 'hidden' and cannot be observed directly; only the emissions which are generated conditionally on the states are observed. From these observations, the most likely state can be inferred. The state process makes HMMs particularly suitable for modeling sequential and temporal data. Thus HMMs have been the state-of-the-art in speech recognition in the past two decades.

In the isolated word recognition case, each spoken word is represented as an observation **O**:

$$\mathbf{O} = o_1, o_2, ..., o_T$$

where o_t represents a observed speech vector at time t. In practice, we often have readily trained models for particular words or phonemes. In this case the word recognition task is to assign the observed speech vector to the model that gives the greatest probability of observing the speech vector:

$$argmax_M P(O|M)$$

3.3 Deep Neural Network Hidden Markov Model (DNN-HMM) hybrid model

The DNNs cannot be used directly for modeling of speech, because it is a time series signal and DNNs require fixed-sized inputs. In order to exploit the strong classification capabilities of DNNs in speech recognition, there is a need to resolve a problem of variable lengths in speech signal. As one of the solutions, in a hybrid DNN-HMM model, the dynamics of the speech signal (sequential properties of speech) are modeled with HMMs and the observation probabilities (emission probabilities modeled with GMM in the traditional GMM-HMM system) are estimated through DNNs. These models outperform [3] traditional Gaussian mixture model (GMM)-HMM on many large vocabulary continuous speech recognition tasks.

This hybrid model has emerged as an alternative ASR method during the end of 1980s and beginning of 1990s, and recently is experiencing a comeback after the DNN's strong representational powers became well known. Here, every output neuron of DNN is trained in order to estimate, based on a given acoustic observation, the posterior probability of continuous density HMMs' state. More specifically, for and observation o_{ut} corresponding to time t in utterance u, the output $y_{ut}(s)$ of DNN to the HMM state s is obtained using the softmax activation function:

$$y_{ut}(s) \triangleq P(s|o_{ut}) = \frac{exp\{a_{ut}(s)\}}{\sum_{s'} exp\{a_{ut}(s')\}},$$

where a_{ut} is the activation at the output layer corresponding to state s. The recognizer uses a pseudo log-likelihood of state s given observation o_{ut},

$$\log p(o_{ut}|s) = \log y_{ut}(s) - \log P(s),$$

where $P(s)$ is the prior probability of state s calculated from training data.

Besides already mentioned advantages, this hybrid model can be trained using the embedded Viterbi algorithm and the decoding is usually very efficient. An example of a DNN-HMM model is shown in figure 2.

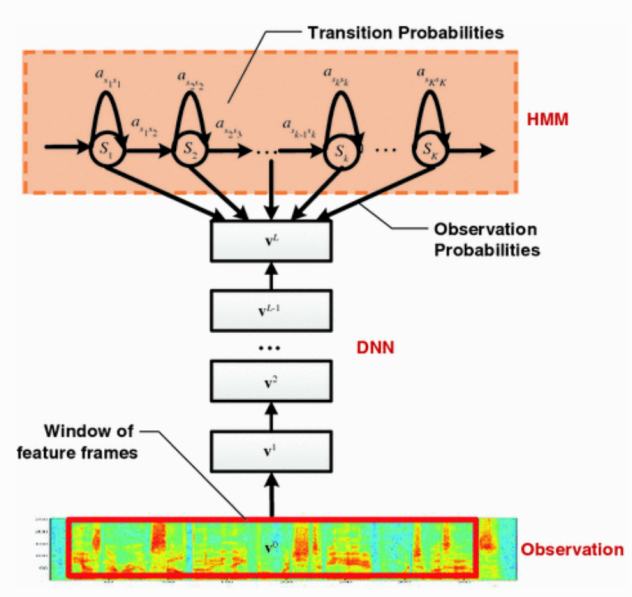

Figure 2: Hybrid DNN-HMM system[3]

4 Experiments

4.1 Data and features

The data used in the following experiments is from the TIDIGITS dataset of recordings[8].The data consists of 326 speakers with 111 adult males, 114 adult females, 50 boys and 51 girls each subject pronounces 77 digit sequences. The data is partitioned between a training and testing set.

Using the Hidden Markov Toolkit HTK [9], we computed Filterbank Features and Mel Frequency Cepstral Coefficients (MFCC) Features with the Deltas (first order differences) and Accelerations (second order differences).

Once the features are calculated we then use a GMM-HMM model and a Vertibi decoder in HTK to align the features to phoneme states, each phoneme having three states, with a single state for short pauses. This creates the ground truth which is required for training the neural networks. Given that the TIDIGTS has 22 possible phonemes there will be 64 possible states and therefore outputs for the DNNs.

Before using the features as inputs to the DDNs, they were normalised on a per dimension basis by subtracting the dimension mean and dividing by the dimension standard deviation. This to reduce the effect of scale in DNN training.

We then reserve 10% of the training data into a validation set maintaining an equal distribution of males and females speakers in the validation set.

4.2 Deep Neural Networks

Using the PDNN package [10] for fitting deep neural networks we used the above mentioned features to train a DNNs with varying architectures and features. The networks were setup as follows:

- Wide Net:with 4 hidden layers each with 1024 nodes and Filterbank features.
- Wide Net:with 4 hidden layers each with 1024 nodes and MFCCs with Deltas and Accelerations.
- Narrow Net:with 4 hidden layers each with 256 nodes and MFCCs with Deltas and Accelerations.
- Wide Net:with 4 hidden layers each with 1024 nodes and MFCCs with Deltas and Accelerations including a context of 5.

A 'Context of 5' is created by augmenting the each frame with five frames that precede it and five frames that come after it. This intern extends the input dimension to 11 times the original dimension.

While training, we obtained training and validation errors. After training, we selected the best network based on a validation error basis and computed the posterior class probabilities on the test set. Using the maximum posterior probability we determined the network's predicted output state (class). After that, we computed the classification accuracy of the posterior predictions for states.

Next, we proceed to aggregate the states to phonemes, such that if the DNN predicts any state of a phoneme, it is considered correct if it matches the ground truth phoneme regardless of whether it matches the target at a state level.

4.3 DDN-HMM for Isolated Word Recognition

Lastly, we constructed a Hybrid DNN-HMM system by replacing the traditional GMM emissions with the outputs of the 'winning' DNN. We do this only on the isolated utterances of the TIDIGITS test set. There are 2486 isolated utterances in the test set.

However, since the DNN gives the posterior probability of the state given the frame $p(s|x)$, we need to convert it to an emission probability $p(x|s)$ in order to use it as an HMM emission. Using the Bayes rule:

$$p(x|s) = \frac{p(s|x)p(x)}{p(s)},$$

we can achieve this by ignoring $p(x)$, which is independent of the state we then obtain the 'scaled posterior' $\tilde{p}(x|s)$.

Then, we create an encoding between the state s_i of the HMM and its corresponding 'scaled posterior' $\tilde{p}(x|s_i)$ by looking up each phoneme in the model pronunciation and matching it to the relevant range in the scaled posterior. The order of states is assumed to be similar in both DNN and HMM.

Further more, we score each of the 11 models(from lab 2) for digits using the forward algorithm in order to pick the model which gives the maximum probability of observing the presented utterance. The digit corresponding to the 'optimal' model is compared to the ground truth to access accuracy of the system.

5 Results

Figure 3 shows the training and validation errors for the various DNNs described in section 4.2. From the charts, it can be seen that the network with context and 1024 hidden neurons outperforms the other networks with a training error of 1.185483% and validation error of 15.92197%. We will refer to this network as the 'best net'.

As described in section 4.2 we calculated the accuracy of state and phoneme classification using maximum posterior probabilities of 'best net' on the test set. We obtained accuracies of 84% at state level and 92% at phoneme level, implying a Phoneme Error Rate(PER) of 8%. Figure 4 shows the confusion matrix using posterior predictions at phoneme level.

We then used the 'best net' to construct a hybrid system as described in section 4. After scoring the different models for each digit using the forward algorithm to compute the predicted word from the hybrid system, we obtained a digit recognition accuracy of 95.09% on the isolated utterances of the test set. Figure 5 shows the confusion matrix for the isolated word recognition using the hybrid system.

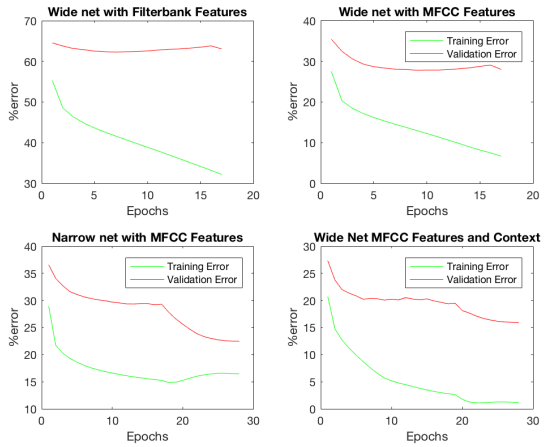

Figure 3: Training and Validation errors obtained when training various DNNs

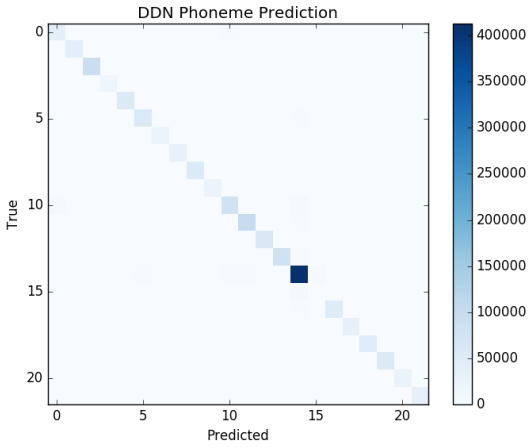

Figure 4: Confusion matrix for 'best net' on testset phonemes

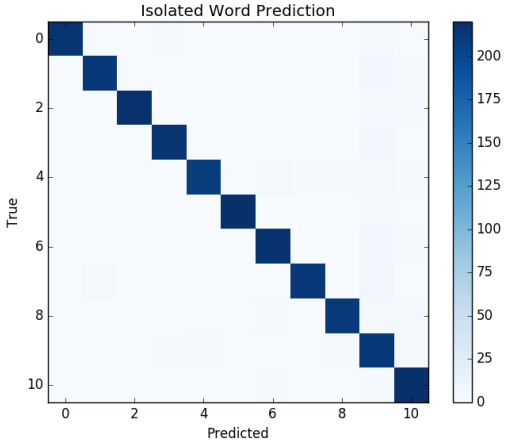

Figure 5: Confusion matrix for the hybrid DNN-HMM system on test set for isolated digits

6 Discussion and Conclusions

We have seen that given the same architecture, DNNs with MFCC features outperform DNNs with filterbank features in the phoneme state prediction task. Further to this we see that adding context also improves phoneme recognition. Trivially, we also observe that wider nets outperform narrow nets with the same number of layers. We believe the above can be explained by the additional information contained in the neighbouring windows in the case of context, and the number of free parameters in the case of network width.

Using the hybrid system we obtain reasonably high accuracies in isolated word recognition. However GMM-HMM models have been previously been found to have similar and even higher accuracies with significantly fewer parameters as in lab 2. We believe the reason for this is that the DNN requires much more data in order to fine tune the vast number of free parameters.

As an addition to this work, like Hinton et.al [2] one can consider a larger set of candidate DNN architectures which might outperform 'best net'. This could also include Convolutional Neural Networks. Pre-processing training data with Deep Belief Networks or Autoencoders has also been shown to improve DNN performance. Another interesting extension would be to compare the hybrid system performance with that of a Recurrent Neural Network.

References

[1] Wikipedia, (2016). *Speech Recognition*. [online] Available at: https://en.wikipedia.org/wiki/Speech$_recognition$

[2] Hinton, G., Deng L., Dong, Y., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Saintah, T. and Kingsbury, B. (2012). *Deep Neural Networks for Acoustic Modelling for Speech Recognition*. IEEE Signal Processing Magazine 29(6):82-97

[3] Dong, Y., Deng, L. (2015). *Automatic Speech Recognition A Deep Learning Approach*. London: Springer, pp. 99 116.

[4] Dahl, G. E., Dong, Y., Deng, L., Acero, A. (2012). *Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition*. IEEE Transactions on Audio, Speech and Language Processing, Vol 20., NO. 1.

[5] Seide, F., Gang, L., Dong, L. (2011). *Conversational Speech Transcription Using Context-Dependent Deep Neural Networks*. Proc. Interspeech.

[6] Mohamed, A., Dahl, G., Hinton, G. (2009). Deep Belief Networks for Phone Recognition. Proc. NIPS Workshop Deep Learn. Speech Recogn. Rel. Applicat.

[7] Cybenko, G. (1989). *Aproximation By Superposition of a Sigmoidal Function*.Mathematics of Control, Signals and Systems.Volume 2. PP 303-314.

[8] R. Gary Leonard and George Doddington. Tidigits ldc93s10. Philadelphia: Linguistic Data Consortium, 1993.

[9] S.J. Young and Sj Young. The htk hidden markov model toolkit: Design and philosophy. Entropic Cambridge Research Laboratory, Ltd, 2:244, 1994.

[10] Yajie Miao. Kaldi+pdnn: Building dnn-based ASR systems with kaldi and PDNN. CoRR, abs/1401.6984, 2014.