UNDERSTANDING HOW DEEP BELIEF NETWORKS PERFORM ACOUSTIC MODELLING

Abdel-rahman Mohamed, Geoffrey Hinton, and Gerald Penn

Department of Computer Science, University of Toronto

ABSTRACT

Deep Belief Networks (DBNs) are a very competitive alternative to Gaussian mixture models for relating states of a hidden Markov model to frames of coefficients derived from the acoustic input. They are competitive for three reasons: DBNs can be fine-tuned as neural networks; DBNs have many non-linear hidden layers; and DBNs are generatively pre-trained. This paper illustrates how each of these three aspects contributes to the DBN's good recognition performance using both phone recognition performance on the TIMIT corpus and a dimensionally reduced visualization of the relationships between the feature vectors learned by the DBNs that preserves the similarity structure of the feature vectors at multiple scales. The same two methods are also used to investigate the most suitable type of input representation for a DBN.

Index Terms— Deep belief networks, neural networks, acoustic modeling

1. INTRODUCTION

Although automatic speech recognition (ASR) has evolved significantly over the past few decades, ASR systems are challenged when they encounter audio signals that differ significantly from the limited conditions under which they were originally trained. The long term research goal is to develop systems that are capable of dealing with the large variety of speech, speaker, channel, and environmental conditions which people typically encounter. Models with high capacity are needed to model this diversity in the speech signal. A typical ASR system uses Hidden Markov Models (HMMs) to model the sequential structure of speech signals, with each HMM state using a Gaussian mixture model (GMM) to model some type of spectral representation of the sound wave. Some ASR systems use feedforward neural networks [1, 2].

DBNs [3] were proposed for acoustic modeling in speech recognition [4] because they have a higher modeling capacity per parameter than GMMs and they also have a fairly efficient training procedure that combines unsupervised generative learning for feature discovery with a subsequent stage of supervised learning that fine-tunes the features to optimize discrimination. Motivated by the good performance of DBNs on the TIMIT corpus, several leading speech research groups have used DBN acoustic models for a variety of LVCSR tasks [5, 6] achieving very competitive performance.

This paper investigates which aspects of the DBN are responsible for its good performance. The next section introduces the evaluation setup that is used throughout the paper. Section 3 discusses the three main strengths of a DBN acoustic model. Then section 4 uses the arguments of section 3 to propose better input features for a DBN.

2. EVALUATION SETUP

We used phone recognition error rates (PER) on the TIMIT corpus to evaluate how variations in the acoustic model influence recognition performance. We removed all SA records (i.e., identical sentences for all speakers in the database) for both training and testing. Some of the SA records are used for the feature visualization experiment in section 4. A development set of 50 speakers was used for tuning the meta-parameters while results are reported using the 24-speaker core test set. The speech was analyzed using a 25-ms Hamming window with a 10-ms fixed frame rate. Three different types of features were used: Fourier-transform-based log filter-bank with 40 coefficients (and energy) distributed on a mel-scale (referred to as "fbank"), dct transformed fbank features ("dct"), and 12th-order Mel frequency cepstral coefficients derived from the dct features ("MFCC"). All features were augmented with their first and second temporal derivatives. Then data were normalized so that each coefficient or first derivative or second derivative had zero mean and unit variance across the training cases. We used 183 target class labels: 3 states for each of the 61 phones. After decoding, the 61 phone classes were mapped to a set of 39 classes for scoring. All of our experiments used a bigram language model over phones, estimated from the training set.

3. ANATOMY OF A DBN ACOUSTIC MODEL

Strictly speaking, a DBN is a graphical model with multiple layers of binary latent variables that can be learned efficiently, one layer at a time, by using an unsupervised learning procedure that maximizes a variational lower bound on the log probablility of the acoustic input. Because of the way it is learned, the graphical model has the convenient property that the top-down generative weights can be used in the opposite direction for performing inference in a single bottom-up pass. This allows the learned graphical model to be treated as a feedforward multi-layer neural network which can then be fine-tuned to optimize discrimination using back-propagation. In a mild abuse of terminology, the resulting feedforward neural network is also called a DBN.

Systems with DBN acoustic models achieve good recognition performance because of three distinct properties of the DBN: it is a neural network which is a very flexible model; it has many non-linear hidden layers which makes it even more flexible; it is generatively pretrained which acts as a strong, domain-dependent regularizer on the weights.

3.1. The advantage of being a neural network

Neural networks offer several potential modelling advantages. First, a neural network's estimation of the HMM state posteriors does not require detailed assumptions about the data distribution. They can also easily combine diverse features, including both discrete and

continuous features. A very important feature of neural networks is their "distributed representation" of the input, i.e., many neurons are active simultaneously to represent each input vector. This makes neural networks exponentially more compact than GMMs. Suppose, for example, that N significantly different patterns can occur in one sub-band and M significantly different patterns can occur in another. Suppose also the patterns occur in each sub-band roughly independently. A GMM model requires NM components to model this structure because each component of the mixture must generate both sub-bands; each piece of data has only a single latent cause. On the other hand, a model that explains the data using multiple causes only requires $N+M$ components, each of which is specific to a particular sub-band. This property allows neural networks to model a diversity of speaking styles and background conditions with much less training data because each neural network parameter is constrained by a much larger fraction of the training data than a GMM parameter.

3.2. The advantage of being deep

The second key idea of DBNs is "being deep." Deep acoustic models are important because the low level, local, characteristics are taken care of using the lower layers while higher-order and highly non-linear statistical structure in the input is modeled by the higher layers. This fits with human speech recognition which appears to use many layers of feature extractors and event detectors [7]. The state-of-the-art ASR systems use a sequence of feature transformations (e.g., LDA, STC, fMLLR, fBMMI), cross model adaptation, and lattice-rescoring which could be seen as carefully hand-designed deep models. Table 1 compares the PERs of a shallow network with one hidden layer of 2048 units modelling 11 frames of MFCCs to a deep network with four hidden layers each containing 512 units. The comparison shows that, for a fixed number of trainable parameters, a deep model is clearly better than a shallow one.

Table 1. *The PER of a shallow and a deep network.*

Model	1 layer of 2048	4 layers of 512
dev	23%	21.9%
core	24.5%	23.6%

3.3. The advantage of generative pre-training

One of the major motivations for generative training is the belief that the discriminations we want to perform are more directly related to the underlying causes of the acoustic data than to the individual elements of the data itself. Assuming that representations that are good for modeling $p(data)$ are likely to use latent variables that are more closely related to the true underlying causes of the data, these representations should also be good for modeling $p(label|data)$. DBNs initialize their weights generatively by layerwise training of each hidden layer to maximize the likelihood of the input from the layer below. Exact maximum likelihood learning is infeasible in networks with large hidden layers because it is exponentially expensive to compute the derivative of the log probability of the training data. Nevertheless, each layer can be trained efficiently using an approximate training procedure called "contrastive divergence" [8]. Training a DBN without the generative pre-training step to model 15 frames of fbank coefficients caused the PER to jump by about 1% as shown in figure(1). We can think of the generative pre-training phase as a strong regularizer that keeps the final parameters close to a good generative model. We can also think of the pre-training as an optimization trick that initializes the parameters near a good local maximum of $p(label|data)$.

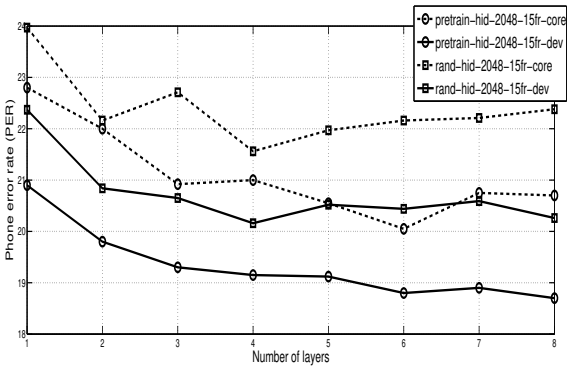

Fig. 1. PER as a function of the number of layers.

4. WHICH FEATURES TO USE WITH DBNS

State-of-the-art ASR systems do not use fbank coefficients as the input representation because they are strongly correlated so modeling them well requires either full covariance Gaussians or a huge number of diagonal Gaussians which is computationally expensive at decoding time. MFCCs offer a more suitable alternative as their individual components tend to be independent so they are much easier to model using a mixture of diagonal covariance Gaussians. DBNs do not require uncorrelated data so we compared the PER of the best performing DBNs trained with MFCCs (using 17 frames as input and 3072 hidden units per layer) and the best performing DBNs trained with fbank features (using 15 frames as input and 2048 hidden units per layer) as in figure 2. The performance of fbank features is about 1.7% better than MFCCs which might be wrongly attributed to the fact that fbank features have more dimensions than MFCCs. Dimensionality of the input is not the crucial property (see p. 3).

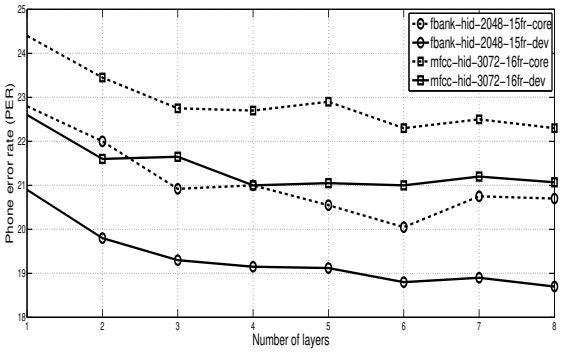

Fig. 2. PER as a function of the number of layers.

To understand this result we need to visualize the input vectors (i.e. a complete window of say 15 frames) as well as the learned hidden activity vectors in each layer for the two systems (DBNs with 8 hidden layers plus a softmax output layer were used for both systems). A recently introduced visualization method called "t-SNE" [9] was used for producing 2-D embeddings of the input vectors or the hidden activity vectors. t-SNE produces 2-D embeddings in which points that are close in the high-dimensional vector space

are also close in the 2-D space. It starts by converting the pairwise distances, d_{ij} in the high-dimensional space to joint probabilities $p_{ij} \propto exp(-d_{ij}^2)$. It then performs an iterative search for corresponding points in the 2-D space which give rise to a similar set of joint probabilities. To cope with the fact that there is much more volume near to a high dimensional point than a low dimensional one, t-SNE computes the joint probability in the 2-D space by using a heavy tailed probability distribution $q_{ij} \propto (1 + d_{ij}^2)^{-1}$. This leads to 2-D maps that exhibit structure at many scales [9].

For visualization only (they were not used for training or testing), we used SA utterances from the TIMIT core test set speakers. These are the two utterances that were spoken by all 24 different speakers. Figures 3 and 4 show visualizations of fbank and MFCC features for 6 speakers. Crosses refer to one utterance and circles refer to the other one, while different colours refer to different speakers. We removed the data points of the other 18 speakers to make the map less cluttered.

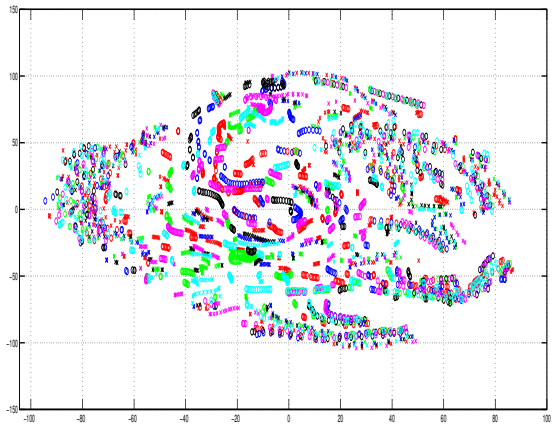

Fig. 3. t-SNE 2-D map of fbank feature vectors

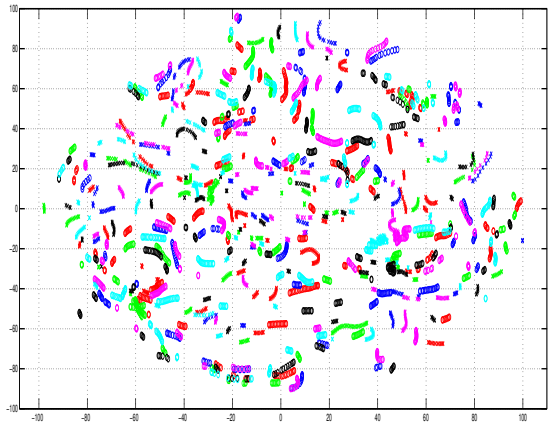

Fig. 4. t-SNE 2-D map of MFCC feature vectors

MFCC vectors tend to be scattered all over the space as they have decorrelated elements while fbank feature vectors have stronger similarities and are often aligned between different speakers for some voiceless sounds (e.g. /s/, /sh/). This suggests that the fbank feature vectors are easier to model generatively as the data have stronger local structure than MFCC vectors. We can also see that DBNs are doing some implicit normalization of feature vectors across different speakers when fbank features are used because they contain both the spoken content and style of the utterance which allows the DBN (because of its distributed representations) to partially separate content and style aspects of the input during the pre-training phase. This makes it easier for the discriminative fine-tuning phase to enhance the propagation of content aspects to higher layers. Figures 5, 6, 7 and 8 show the 1st and 8th layer features of fine-tuned DBNs trained with fbank and MFCC respectively. As we go higher in the network, hidden activity vectors from different speakers for the same segment align in both the MFCC and fbank cases but the alignment is stronger in the fbank case.

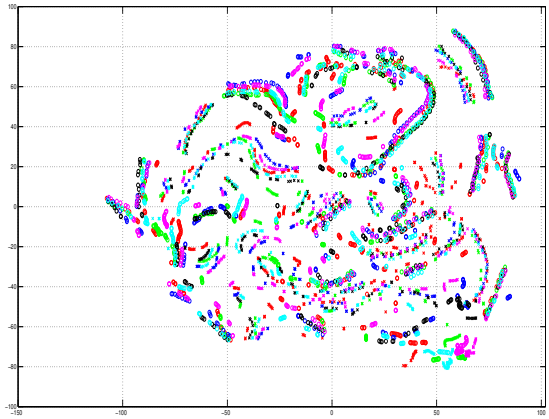

Fig. 5. t-SNE 2-D map of the 1st layer of the fine-tuned hidden activity vectors using fbank inputs.

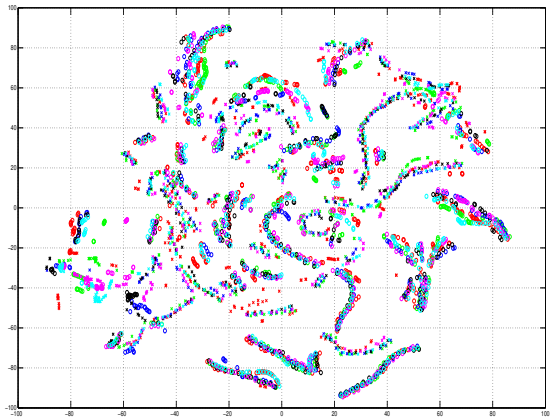

Fig. 6. t-SNE 2-D map of the 8th layer of the fine-tuned hidden activity vectors using fbank inputs.

To refute the hypothesis that fbank features yield lower PER because of their higher dimensionality, we consider dct features, which are the same as fbank features except that they are trans-

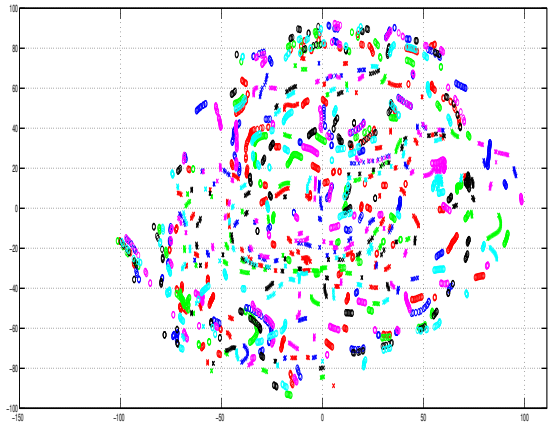

Fig. 7. t-SNE 2-D map of the 1st layer of the fine-tuned hidden activity vectors using MFCC inputs.

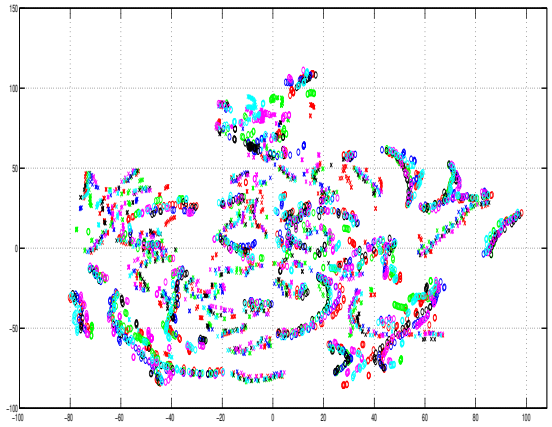

Fig. 8. t-SNE 2-D map of the 8th layer of the fine-tuned hidden activity vectors using MFCC inputs.

formed using the discrete cosine transform, which encourages decorrelated elements. We rank-order the dct features from lower-order (slow-moving) features to higher-order ones. For the generative pre-training phase, the dct features are disadvantaged because they are not as strongly structured as the fbank features. To avoid a confounding effect, we skipped pre-training and performed the comparison using only the fine-tuning from random initial weights. Table 2 shows PER for fbank, dct, and MFCC inputs (11 input frames and 1024 hidden units per layer) in 1, 2, and 3 hidden-layer neural networks. dct features are worse than both fbank features and MFCC features. This prompts us to ask why a lossless transformation causes the input representation to perform worse (even when we skip a generative pre-training step that favours more structured input), and how dct features can be worse than MFCC features, which are a subset of them. We believe the answer is that higher-order dct features are useless and distracting because all the important information is concentrated in the first few features. In the fbank case the discriminant information is distributed across all coefficients. We conclude that the DBN has difficulty ignoring irrelevant input features. To test this claim, we padded the MFCC vector with random noise to be of the same dimensionality as the dct features and then used them for network training (MFCC+noise row in table 2). The MFCC performance was degraded by padding with noise. So it is not the higher dimensionality that matters but rather how the discriminant information is distributed over these dimensions.

Table 2. *The PER deep nets using different features*

Feature	Dim	1lay	2lay	3lay
fbank	123	23.5%	22.6%	22.7%
dct	123	26.0%	23.8%	24.6%
MFCC	39	24.3%	23.7%	23.8%
MFCC+noise	123	26.3%	24.3%	25.1%

5. CONCLUSIONS

A DBN acoustic model has three main properties: It is a neural network, it has many layers of non-linear features, and it is pre-trained as a generative model. In this paper we investigated how each of these three properties contributes to good phone recognition on TIMIT. Additionally, we examined different types of input representation for DBNs by comparing recognition rates and also by visualising the similarity structure of the input vectors and the hidden activity vectors. We concluded that log filter-bank features are the most suitable for DBNs because they better utilize the ability of the neural net to discover higher-order structure in the input data.

6. REFERENCES

[1] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, 1993.

[2] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *ICASSP*, 2000, pp. 1635–1638.

[3] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[4] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[5] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[6] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *ASRU*, 2011.

[7] J.B. Allen, "How do humans process and recognize speech?," *IEEE Trans. Speech Audio Processing*, vol. 2, no. 4, pp. 567–577, 1994.

[8] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1711–1800, 2002.

[9] L.J.P. van der Maaten and G.E. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.