LEARNING A BETTER REPRESENTATION OF SPEECH SOUND WAVES USING RESTRICTED BOLTZMANN MACHINES

Navdeep Jaitly, Geoffrey Hinton

Department of Computer Science, University of Toronto, Toronto, M5S 3G4, Canada

ABSTRACT

State of the art speech recognition systems rely on pre-processed speech features such as Mel cepstrum or linear predictive coding coefficients that collapse high dimensional speech sound waves into low dimensional encodings. While these have been successfully applied in speech recognition systems, such low dimensional encodings may lose some relevant information and express other information in a way that makes it difficult to use for discrimination. Higher dimensional encodings could both improve performance in recognition tasks, and also be applied to speech synthesis by better modeling the statistical structure of the sound waves. In this paper we present a novel approach for modeling speech sound waves using a Restricted Boltzmann machine (RBM) with a novel type of hidden variable and we report initial results demonstrating phoneme recognition performance better than the current state-of-the-art for methods based on Mel cepstrum coefficients.

Index Terms— Restricted Boltzmann Machine, RBM, phoneme recognition, TIMIT

1. INTRODUCTION

Speech sound waves have complex distributions that have long evaded statistical modeling at the level of raw signals. To avoid these complexities automated speech recognition systems typically use low-dimensional speech encodings such as Mel frequency scale cepstral coefficients (MFCC), linear predictive coding (LPC) or perceptual linear prediction (PLP). With the advent of new machine learning algorithms such as Independent Components Analysis [1] some progress has been made in learning to extract features directly from the sound wave [2, 3]. There is, however, room for further progress in developing more refined generative models of raw speech that can be used both for generating speech and for tasks such as automated speech recognition [4].

Restricted Boltzmann Machines (RBMs) are undirected graphical models that use hidden/latent variables in energy based models to achieve highly expressive marginal distributions [5]. Although maximum likelihood learning is intractable in these models, the Contrastive Divergence (CD) algorithm [6] has been shown to be very effective in training RBMs to model a variety of high dimensional data distributions such as images and image transformations [7, 8]. Several RBMs can be stacked on top of each other such that higher level RBMs learn to model the posterior distributions of the hidden variables of the lower level RBMs. This stacking process has the property that, under certain conditions, adding another RBM to the stack creates a new composite model, called a Deep Belief Net (DBN) that has a better lower bound on the log probability of the training data than the previous DBN. DBN's trained on MFCCs [9] or Mel scale filter banks [10] create high-level features that can be used to predict a posterior distribution over the states of an HMM, and after fine-tuning with backpropagation, these multilayer neural networks outperform all other speaker-independent methods for recognizing phones on the TIMIT database. Similarly DBN's have been trained on spectrograms [11] and applied to several audio classification tasks.

In this paper we use an RBM to model raw speech signals and show that it can be trained effectively using the CD algorithm. By better capturing the statistics of raw signals we hope to learn features that are more relevant to recognition tasks than the traditional features such as mel filter banks. Indeed, we show that the detected features can be used to achieve better performance in phoneme recognition on the TIMIT corpus [1] than most of the state of the art speaker-independent systems built on mel filter banks and MFCCs.

2. PROBABILISTIC MODEL

We start by briefly describing our model in this section and then describe the algorithm for learning the parameters of the model in the next section (see [7] for a detailed description of the model).

Let $\mathbf{v}$ represent a fragment of speech signal of length 100 samples representing 6.25ms at 16KHz. We model the probability distribution of such fragments on the TIMIT corpus using an RBM which is an energy based model that has hidden (latent) variables $\mathbf{h}$, and visible (observed) variables, $\mathbf{v}$. Each joint configuration of these variables is assigned an en-

Thanks to NSERC, CIFAR & Microsoft for funding.

[1] (`http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1`)

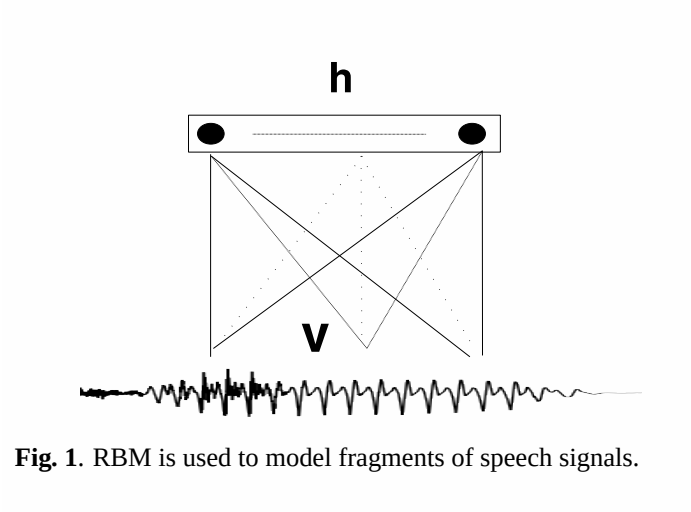

Fig. 1. RBM is used to model fragments of speech signals.

ergy, $E(\mathbf{v}, \mathbf{h})$, and the probabilities of joint configurations are defined by a Boltzmann distribution:

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})} \quad (1)$$

where Z is the partition function $Z = \sum_{(\mathbf{v},\mathbf{h})} \exp(-E(\mathbf{v}, \mathbf{h}))$.

For a Gaussian-Bernoulli RBM, where the visible variables, $\mathbf{v}$, are linear, and the hidden variables, $\mathbf{h}$, are binary, the energy of a joint configuration is:

$$E(\mathbf{v}, \mathbf{h}; \mathbf{\Theta}) = -\mathbf{v^T W h} + \frac{1}{\sigma^2} \parallel \mathbf{v} - \mathbf{b} \parallel^2 - \mathbf{h^T a} \quad (2)$$

where, the matrix $\mathbf{W}$ represents interaction strengths between different visible variables (rows) and hidden variables (columns). $\mathbf{b}$ and $\mathbf{a}$ represent the visible and hidden biases, and $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{b}, \mathbf{a}\}$[2]. The conditional distribution of the visible variables, given the hidden variables is the Gaussian $N(\sigma^2 \mathbf{W h} + \mathbf{b}; \sigma^2 \mathbf{I})$.

It is possible to replace an individual binary hidden unit by an infinite number of binary variables coupled together with the same incoming weights but with biases stepped downwards by 1 (starting at -0.5). It was shown in [7] that such a coupled set of binary variables, called Stepped Sigmoid Units (SSUs), has the property that the distribution of the sum of activities of the coupled units can be closely approximated by a rectified linear unit whose value is $max(0, N(x, \sigma(x))$, where x represents the input into each of the binary variables (without addition of any bias) and $\sigma(x)$ is the logistic sigmoid function. In this paper, we use this approximation to train a Gaussian-SSU RBM to model the raw speech signal.

3. LEARNING

The gradient of the log of the probability of the training data w.r.t. a weight parameter, $w_{ij} \in \mathbf{W}$ is as follows [6]:

$$\frac{\partial}{\partial w_{ij}} \log p(\mathbf{v}; \mathbf{\Theta}) = < v_i h_j >_{data} - < v_i h_j >_{model} \quad (3)$$

[2]Some authors use a parameterization in which the weight matrix is $\sigma \mathbf{W}$

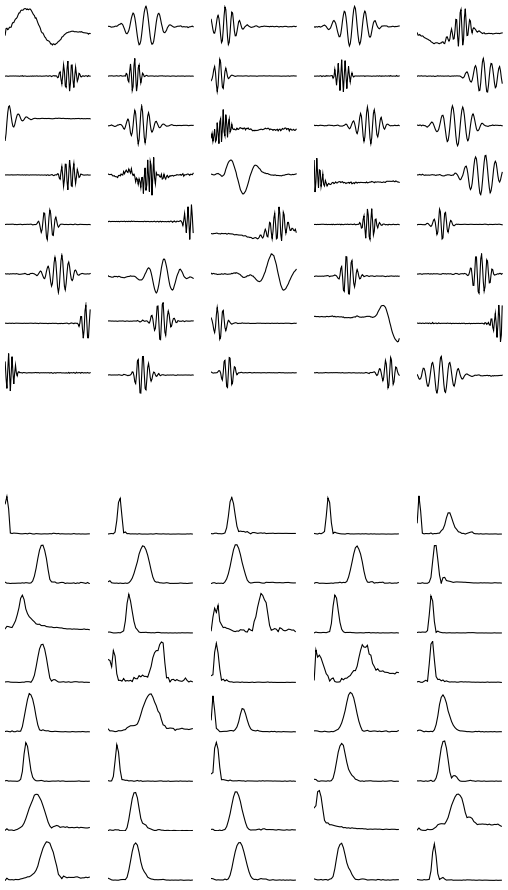

Fig. 2. 40 randomly selected features (top), and DFT of corresponding features (bottom). Features with a bimodal DFT may be features that are modeling vowels.

where $<>_{data}$ denotes the expectation under the distribution of $\mathbf{h}$, conditioned on the data vector, $\mathbf{v}$, and $<>_{model}$ denotes the expectation under the models' distribution over all joint configurations $(\mathbf{v}, \mathbf{h})$.

The first term in equation 3 is easy to compute because the hidden units are conditionally independent given a visible vector. The second term is exponentially expensive to compute exactly, but a crude and very cheap approximation called "Contrastive Divergence" (CD) works surprisingly well for learning [6]. For each training vector, $\mathbf{v}$, in a randomly selected mini-batch, we sample the sum of activities of the SSUs conditioned on $\mathbf{v}$ using the rectified linear approximation mentioned above. Then we resample the visible states from their distribution conditioned on the hidden states to get a "reconstruction" $\mathbf{v}'$. Finally we resample the hidden states conditioned on $\mathbf{v}'$ to get $\mathbf{h}'$, and use the resampled visible and hidden states in the second term of equation 3 instead of using the correct expectations. The biases are learned in the same way by simply omitting either the visible or the hidden

# hiddens in NN	window width & stride (ms)	include Δ,$\Delta\Delta$	# hiddens in RBM	PER(%)	
				validation	test
1000	10:5	No	120	25.8	
2000				23.9	
3000				23.7	
2000	10:5	No	120	23.9	
	25:10			24.9	
2000	25:10	No	120	24.9	
		Yes		23.2	
2000	25:10	Yes	80	25.9	
			120	23.2	24.9
			160	23.4	
4000	10:5	Yes	120	21.5	22.8

Table 1. Table showing results of different settings on PER using only two hidden layers

states from equation 3. In addition, for this paper, we learnt the parameter, σ, by using the difference $\partial E(\mathbf{v}', \mathbf{h}')/\partial\sigma$ - $\partial E(\mathbf{v}, \mathbf{h})/\partial\sigma$ with a very small learning rate.

4. EXPERIMENTS

We trained the above model using training data from the TIMIT database. A Graphical Processing Unit (GPU) in an NVIDIA Tesla S1070 system was used to perform the bulk of the computations (matrix multiplications and sampling) using the Cudamat library [12].

4.1. RBM Training

The entire TIMIT training dataset was normalized to have a standard deviation of 10. Normalization is essential to prevent saturation of hidden units, which produces small learning signal. A standard deviation of 10 was found to give lower percentage reconstruction error than other values, with the parameter initializations we were using. An RBM with 100 Gaussian visibles and 120 SSU hiddens was initialized with weights drawn from $N(0, .01^2)$ and hidden biases drawn from $p(a) = 2\exp(-2a)$. Stochastic gradient descent was performed using mini-batches of 100 randomly selected windows of speech (which were thus in any random phase with respect to the start of the parent sentences). We used a momentum of 0.5 and a learning rate of 10^{-4} (see [13] for details). The training time was chosen so that each point in the training set was sampled 30 times, on average. Figure 2 shows 40 features selected at random and their fourier transforms.

4.2. Phoneme Recognition using the Features

A baseline model with 61 mono-phone HMMs with 3 states for each phoneme was created for the TIMIT database as described in [9]. 'Correct" labels for each 10ms segment of speech in the TIMIT database was found by a forced alignment to this model. We trained a neural network to predict the posterior probabilities over the phoneme labels from the forced aligment. For this we created input for the neural network from the inputs to the features learnt above, as follows. Each sentence was first converted into a series of frames, with consecutive frames overlapping by 99 samples. For each feature, the absolute value of input from contiguous frames were averaged over a window that corresponded to 10ms of speech. Such averaged frames were created starting at every consecutive 5ms. 24 frames (corresponding to a total signal length of 125ms) were concatenated and used as input to predict the phoneme label of the middle frame. Each dimension of the input vector to the neural network was log transformed (values of 0 were replaced with small values) and standardized to have a mean of 0 and a standard deviation of 1. Alternative settings of the subsampling width and frame-advance stride were also considered (see below).

Averages of absolute inputs to the SSU features rather than activations of the features were used because they were seen to outperform the latter. Since the presence of a pattern is more important than the sign of the pattern in the speech signal, this is not surprising. Averaging is beneficial because the encoding produced is not affected by phase, and because averaging enhances the S/N of the signal. Instead of averaging of features values (input or rectified), an alternative would have been to concatenate the feature activations at different frames into one large vector. However, this would create a very high dimensional input to the neural network. Averaging features keeps the number of visible units down to an amenable value.

A two hidden-layer, feed-forward neural network with a "softmax" output layer was trained with back-propagation to predict 183 phone labels. Stochastic gradient descent was used with a mini-batch size of 200 randomly chosen training cases. The parameters at the end of each epoch were used to compute phoneme label probabilities on the development set and these probabilities were decoded using a bigram language model to get a phoneme error rate (PER). The parameters which resulted in the lowest PER on the development set were used to compute the PER on the test set.
We conducted experiments to determine the architecture of the neural network, and preprocessing of the features that optimized the PER on the development set. Table 1 shows the effect of varying the number of hidden units in each layer of the feed-forward neural network, the window width and stride, the presence of Δ and $\Delta\Delta$ features [3] the number of hidden features in the RBM. Without Δ features, smaller subsampling windows of 10 ms with a stride of 5ms appear to perform better than the larger windows. With Δ features, however, the input vector for the discriminative neural network has 8280 components and more hidden nodes (4000) are needed to learn a good function. Using a neural network with three hidden layers that was pretrained as a Deep Belief Net results in a PER of 20.6% on the validation set and 21.8% on the test-set. Table 2 shows a comparison of the reported results from several methods.

[3] Δ and $\Delta\Delta$ are computed from the first and second derivative, repectively of values of features, with respect to time

Method	PER
Large-Margin GMM[14]	30.1%
CD-HMM[15]	27.3%
Augmented Conditional Random Fields[15]	26.6%
Recurrent Neural Nets[16]	26.1%
Bayesian Triphone HMM[17]	25.6%
Monophone HTMs[18]	24.8%
Heterogenous Classifiers[19]	24.4%
Deep Belief Networks (bounding silences ignored)[9]	23.0%
DBN using RBM on Raw Speech (this work)	**21.8%**
DBN using mean covariance RBM on mel filter banks	20.5%

Table 2. Reported accuraccy of different methods.

5. CONCLUSIONS AND FUTURE WORK

We have described how to learn an undirected generative model of the distribution of short segments of the speech sound wave. The model learns interesting features and when these features are used for phoneme recognition on the TIMIT benchmark, they already give results that are better than the state of the art methods using MFCCs, even though we are using the features in a rather naive way. There are many promising variations of our approach that have yet to be tried and we anticipate significant improvements in recognition rates from some of these variations. We are also exploring the use of our features for speech generation and single-source speaker separation.

6. REFERENCES

[1] A.J. Bell and T.J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *NEURAL COMPUTATION*, vol. 7, pp. 1129–1159, 1995.

[2] M. S. Lewicki, "Efficient coding of natural sounds," *Nature Neuroscience*, vol. 5, no. 4, pp. 356–363, April 2002.

[3] J. Lee, H. Jung, T. Lee, and S. Lee, "Speech feature extraction using independent component analysis," 2000, vol. 3, pp. 1631 –1634 vol.3.

[4] M. S. Lewicki, "Information theory: A signal take on speech," *Nature*, vol. 466, no. 7308, pp. 821–822, August 2010.

[5] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel Distributed Processing: Volume 1: Foundations*, D. E. Rumelhart, J. L. McClelland, et al., Eds., pp. 194–281. MIT Press, Cambridge, 1986.

[6] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, pp. 2002, 2000.

[7] V. Nair and G.E. Hinton, "Rectified linear units improve Restricted Boltzmann machines," in *ICML*, 2010, pp. 807–814.

[8] R. Memisevic and G.E. Hinton, "Learning to represent spatial transformations with factored higher-order Boltzmann machines," *Neural Computation*, vol. 22, no. 6, pp. 1473–1492, June 2010.

[9] A. Mohamed, G.E. Dahl, and G. E. Hinton, "Deep belief networks for phone recognition," in *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[10] A. Mohamed G. Dahl, M. Ranzato and G. Hinton, "Phone recognition with the mean-covariance restricted boltzmann machine," in *Advances in Neural Information Processing Systems 23, NIPS'10*, 2010, vol. 23.

[11] P. Pham H. Lee, Y. Largman and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in Neural Information Processing Systems 22, NIPS'09*, 2009, vol. 22.

[12] V. Mnih, "Cudamat: a cuda-based matrix class for python," Tech. Rep. 004, Department of Computer Science, University of Toronto, 2009.

[13] G. Hinton, "A practical guide to training Restricted Boltzmann Machines," Tech. Rep. 003, Department of Computer Science, University of Toronto, 2010.

[14] F. Sha and L.K. Saul, "Large margin gaussian mixture modeling for phonetic classification and recognition," may. 2006, vol. 1, pp. I –I.

[15] Y. Hifny and S. Renals, "Speech recognition using augmented conditional random fields," *Trans. Audio, Speech and Lang. Proc.*, vol. 17, no. 2, pp. 354–365, 2009.

[16] T. Robinson, "An application of recurrent nets to phone probability estimation," *IEEE Transactions on Neural Networks*, vol. 5, pp. 298–305, 1994.

[17] J. Ming and F.J. Smith, "Improved phone recognition using bayesian triphone models," may. 1998, vol. 1, pp. 409 –412 vol.1.

[18] L. Deng and D. Yu, "Use of differential cepstra as acoustic features in hidden trajectory modeling for phonetic recognition," apr. 2007, vol. 4, pp. IV–445 –IV–448.

[19] A.K. Halberstadt and J.R. Glass, "Heterogeneous measurements and multiple classifiers for speech recognition," 1998, vol. 1, p. 0396.