SEQUENTIAL DEEP BELIEF NETWORKS

Galen Andrew

University of Washington
Department of Computer Science
galen@cs.washington.edu

Jeff Bilmes

University of Washington
Department of Electrical Engineering
bilmes@ee.washington.edu

ABSTRACT

Previous work applying Deep Belief Networks (DBNs) to problems in speech processing has combined the output of a DBN trained over a sliding window of input with an HMM or CRF to model linear-chain dependencies in the output. We describe a new model called Sequential DBN (SDBN) that uses inherently sequential models in all hidden layers as well as in the output layer, so the latent variables can potentially model long-range phenomena. The model introduces minimal computational overhead compared to other DBN approaches to sequential labeling, and achieves comparable performance with a much smaller model (in terms of number of parameters). Experiments on TIMIT phone recognition show that including sequential information at all layers improves accuracy over baseline models that do not use sequential information in the hidden layers.

***Index Terms*—** deep learning, deep belief network, phone recognition, TIMIT

1. INTRODUCTION

In Conditional Random Field models for sequential labeling (CRFs) [1], a Markov Random Field (MRF) is defined over a label sequence whose distribution depends on the input. CRFs allow features of the input to be defined at arbitrary distance from the associated label, but the user must consciously design such features to allow long-distance dependencies. Feature design is a difficult, task-specific problem, and it is especially difficult to design effective long-range features for tasks such as speech recognition, where the input is a relatively low-level representation of the acoustic signal. On the other hand, results in speech science suggest that longer-range features (that is, longer than the typical 25ms frame width) may be useful for speech perception, particularly in noisy environments [2].

Several methods have been proposed to introduce hidden variables to CRFs that might be capable of modeling regularities in the data that are not explicit in the features but nevertheless aid in classification. The hidden CRF (HCRF) appends a multinomial hidden state to each phone class and optimizes the marginal likelihood [3], so that subclasses may be induced that are easier to recognize than the original classes. Another successful approach models each phone as a sequence of three sub-phones, the boundaries of which are latent [4]. Other work uses a multi-layer CRF in which the data is mapped through various layers of multinomial sequences that may be either order-1, or order-0 Markov (that is, conditionally independent given the input) [5]. All of these approaches are more effective than a CRF with no latent structure, but better results have been obtained with a richer latent representation, just as has been found to be the case for static classification problems with Deep Belief Networks (DBNs).

Deep Belief Networks [6] have emerged as an empirically effective model for inducing rich feature representations of static (non-sequential) data. Each layer of latent representation is learned by training a Restricted Boltzmann Machine (RBM) to model the data distribution at the next lower layer, using e.g., Contrastive Divergence (CD). Since an RBM has bipartite structure, the hidden variables are independent when conditioned on the input. The vector of expected values of the hidden variables given the input can then be used as the representation for further processing. Typically after training several stacked RBMs in sequence, a discriminative classifier is trained using the final layer as the input, and the parameters of the entire chain of feature transformations are then fine-tuned according to a discriminative training criterion.

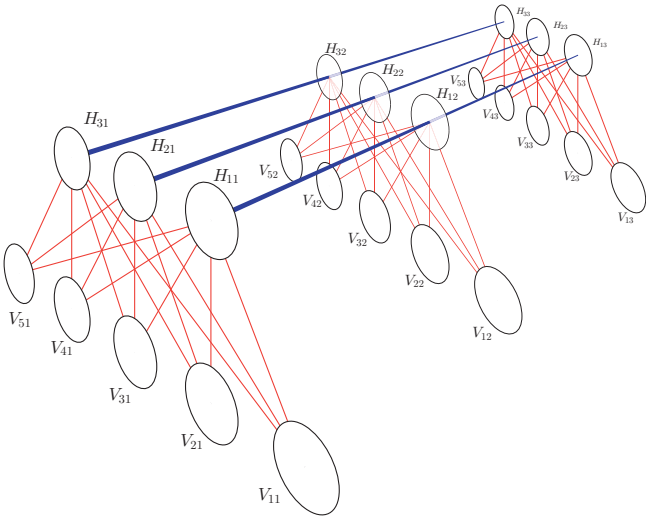
Several researchers have employed DBNs to learn feature representations for use in phone recognition. In [7] a DBN is trained to classify subphones which is then combined with an HMM bigram language model over subphones. In [8], a DBN phone classifier is trained jointly with a CRF that uses the final hidden layer of the DBN as features. The model surpassed the state-of-the-art on the TIMIT phone recognition task, even though due to the static training of the DBN, a) the hidden variables integrate information over a fixed 11 frame (≈110 ms) window and b) nothing can encourage the DBN hidden states to exhibit continuity through time.

In the present work, we use a novel structure called a sequential RBM (SRBM) that allows dependencies between corresponding hidden units at adjacent time frames in the hidden layer. Exact sampling of hidden structures given the input and computation of conditional expectations remains tractable in the SRBM—it involves only matrix-multiplication and traditional forward-backward computations—so CD training is still possible. As with RBMs, we can stack SRBMs and append a sequence classifier (a CRF) to the top layer. The intention is to let the model enforce smoothness in the hidden layers across time-frames, and to allow the hidden variables to potentially model longer-range phenomena. Finally, using a back-propagation-like algorithm, we can discriminatively and jointly fine-tune the parameters of all of the layers. We call the resulting model a Sequential DBN (SDBN).

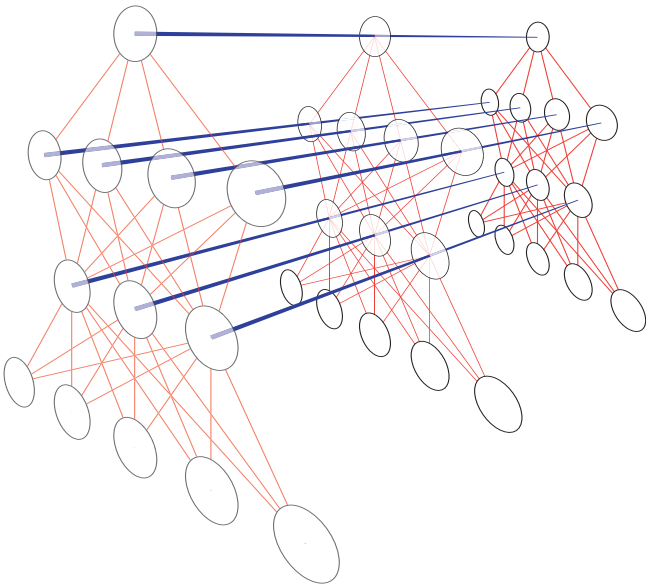
Notation. If X is a matrix, the $(i,j)^{\text{th}}$ entry is X_{ij}, the i^{th} row of X is X_{i*} and the j^{th} column is X_{*j}. The submatrix of columns j through k is $X_{*(j:k)}$. The matrix transpose is denoted X'. If X and Y are matrices (or vectors) of the same dimension, $\langle X, Y\rangle$ denotes $\mathrm{tr}(X'Y)$. If X and Y have the same number of rows, $[X|Y]$ denotes their concatenation. Finally, if $x \in \mathbb{R}^d$ is a vector, $\mathrm{diag}\, x \in \mathbb{R}^{d\times d}$ is the diagonal matrix with x on its diagonal.

2. SEQUENTIAL RESTRICTED BOLTZMANN MACHINES

An SRBM defines a joint distribution over two matrix-valued layers, a visible layer $V \in \mathbb{R}^{n_v \times T}$ and a hidden layer $H \in \mathbb{R}^{n_h \times T}$. As in an RBM, conditioned on the hidden layer, all variables of the visible layer are independent. Conditioned on the visible layer, however,



(a) SRBM



(b) SDBN

Fig. 1. Illustrations of an SRBM and an SDBN. Both consist of $T = 3$ time frames, and have $n_1 = 5$ input units and $n_2 = 3$ hidden units per frame in the first layer. The SDBN has $n_3 = 4$ hidden units per frame in the second layer, plus a single multinomial output per frame. The red edges correspond to the weights of the matrix W_0, while the blue edges have weights given by $\mathbf{t}$. Edges across layers between adjacent time frames corresponding to W_δ (e.g., from V_{11} to H_{12}) are omitted from the figures for clarity.

all *rows* of the hidden layer are independent, but we allow Markov interactions in rows. Allowing temporal dependencies between variables within each row of the hidden layer lets the SRBM potentially model long-range dependencies between the visible layer at widely separated time frames, while retaining the tractability of important operations like marginalizing and sampling.

While an RBM typically has dense connections between the visible and hidden layers, an SRBM has only edges that are local in time. Specifically, we use edges between V_{it} and $H_{j(t+\delta)}$ for all i, j, t and for $|\delta| \leq \delta_{\max}$. The weights on the edges are summarized in the matrices $W_\delta \in \mathbb{R}^{n_v \times n_h}$, where $(W_\delta)_{ij}$ is the weight on all edges $(V_{it}, H_{j(t+\delta)})$. The hidden layer of the SRBM also has a vector of transition parameters $\mathbf{t} \in \mathbb{R}^{n_h}$ that govern the interactions between adjacent frames within each row of H, as we will describe shortly. We intentionally disallow edges between observed units, in order to encourage the hidden layer to model any dependencies between time frames of the observations. Figure 1(a) illustrates the graphical (MRF) structure of an SRBM.[1]

In this work, we assume the hidden variables are always binary,[2] meaning $H \in \{\pm 1\}^{n_h \times T}$, and the observed variables are either binary ($V \in \{\pm 1\}^{n_v \times T}$) or real-valued Gaussian ($V \in \mathbb{R}^{n_v \times T}$). For $\delta_{\max} = 1$, the energy of a configuration is defined in terms of the matrix $A^h \in \mathbb{R}^{n_h \times T}$:

$$A^h = \left[W'_{-1} V_{*(2:T)} \middle| \mathbf{0}\right] + W'_0 V + \left[\mathbf{0} \middle| W'_1 V_{*(1:T-1)}\right]. \quad (1)$$

In (1), the middle term $W'_0 V$ produces the matrix of inputs to each hidden unit coming from the visible units at the same time frame. The other two terms add the influence of visible units at the preceding and subsequent frame. The generalization to $\delta_{\max} > 1$ is straightforward.

Let $U = \operatorname{diag} \mathbf{t}$. If both layers are binary, then $\Pr(V, H) \propto \exp -E_{\mathcal{B}}(V, H)$ where the energy function is

$$E_{\mathcal{B}}(V, H) = -\langle H, A^h \rangle - \sum_{t=1}^{T-1} \langle H_{*t}, U H_{*(t+1)} \rangle. \quad (2)$$

Defining

$$A^v = \left[\mathbf{0} \middle| W_{-1} H_{*(1:T-1)}\right] + W_0 H + \left[W_1 H_{*(2:T)} \middle| \mathbf{0}\right],$$

note that

$$\Pr(V|H) \propto \exp -E_{\mathcal{B}}(V, H) \propto \exp \langle H, A^h \rangle = \exp \langle V, A^v \rangle,$$

so the V_{it} are independent given H, with $\Pr(V_{it}|H) \propto \exp A^v_{it} V_{it}$, or $\Pr(V_{it}|H) = \sigma(2 V_{it} A^v_{it})$ where $\sigma(x) = (1 + \exp -x)^{-1}$.

If the visible layer is Gaussian, then the joint density is $f(V, H) \propto \exp -E_{\mathcal{G}}(V, H)$ where

$$E_{\mathcal{G}}(V, H) = -\langle H, A^h \rangle - \sum_{t=1}^{T-1} \langle H_{*t}, U H_{*(t+1)} \rangle + \frac{1}{2} \langle V, V \rangle.$$

Now $f(V|H) \propto \exp\left(\langle V, A^v \rangle - \frac{1}{2} \sum_{it} V_{it}^2\right)$, so the V_{it} are independent given H, with $V_{it}|H \sim \mathcal{N}(A^v_{it}, 1)$.

Regardless of the type of visible layer, $\Pr(H|V)$ factorizes into terms involving individual H_{jt} and terms involving $H_{jt} H_{j(t+1)}$:

$$\begin{aligned} \Pr(H|V) &\propto \exp\left(\langle H, A^h \rangle + \sum_{t=1}^{T-1} \langle H_{*t}, U H_{*(t+1)} \rangle\right) \\ &= \prod_{j=1}^{n^h} \exp\left(\langle H_{j*}, A^h_{j*} \rangle + \sum_{t=1}^{T-1} \mathbf{t}_j H_{jt} H_{j(t+1)}\right) \quad (3) \\ &= \prod_{j=1}^{n^h} \left(\prod_{t=1}^{T} \exp H_{jt} A^h_{jt}\right) \cdot \left(\prod_{t=1}^{T-1} \exp \mathbf{t}_j H_{jt} H_{j(t+1)}\right), \end{aligned}$$

[1] In the experiments we also include three vectors of bias terms: one for H_{*1}, one for H_{*T} and one that is shared by all columns of H. We omit these from the exposition to keep the formulas uncluttered.

[2] Usually, "binary units" take the values 0/1, but we found that our model can be described more cleanly in terms of ± 1 units. Given the way activation is defined in (3), using ± 1 units corresponds to the use of tanh activation functions in an MLP with standard 0/1 units.

So given V, the rows of H are independent order-1 Markov sequences with binary states, and the forward-backward algorithm can be used to sample from $\Pr(H|V)$ and to determine $\mathbb{E}[H|V]$.

It is not hard to show that the gradient of the log-likelihood $\log \Pr(V = \hat{V})$ with respect to the W_δ has the following form, similar to a standard RBM:

$$\nabla_{W_0} = \hat{V}\Big(\mathbb{E}\big[H' \mid V = \hat{V}\big] - \mathbb{E}\big[H'\big]\Big)$$
$$\nabla_{W_1} = \hat{V}_{*(1:T-1)}\Big(\mathbb{E}\big[H'_{*(2:T)} \,\big|\, V = \hat{V}\big] - \mathbb{E}\big[H'_{*(2:T)}\big]\Big)$$

Also, the gradient with respect to $\mathbf{t}_j$ is

$$\nabla_{\mathbf{t}_j} = \sum_{t=1}^{T-1}\Big(\mathbb{E}\big[H_{jt}H_{j(t+1)} \mid V = \hat{V}\big] - \mathbb{E}\big[H_{jt}H_{j(t+1)}\big]\Big).$$

The positive terms (the conditional expectations) can all be computed exactly by first computing the values $\mathbb{E}[H_{jt}|\hat{V}]$ and $\mathbb{E}[H_{jt}H_{j(t+1)}|\hat{V}]$ with Baum-Welch. To approximate the negative terms, we sample $\tilde{V}$ by running two steps of blocked Gibbs sampling, from $\hat{V}$ to H and back, and then use the conditional expectations given $\tilde{V}$, which is analogous to CD training for an RBM.

3. THE SEQUENTIAL DEEP BELIEF NETWORK

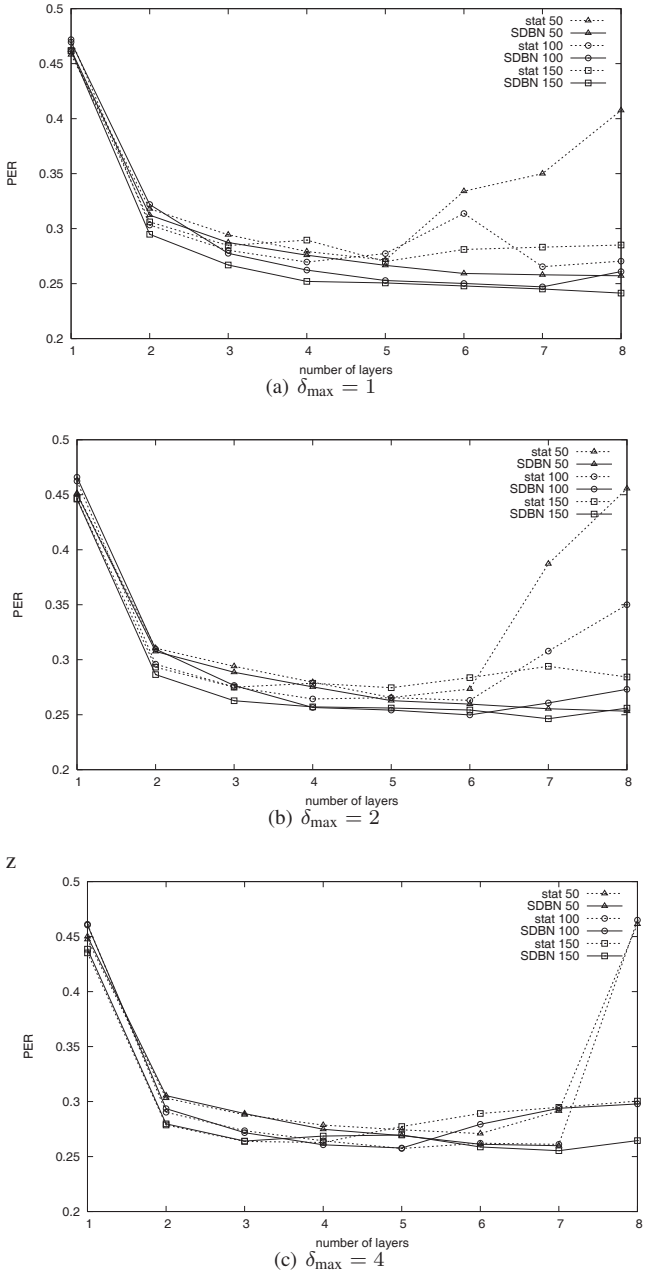
An L-layer SDBN is formed by stacking multiple layers of SRBMs. For $l = 1 \ldots L-1$, the hidden layer at level l is a binary matrix $H^l \in \{\pm 1\}^{n_l \times T}$ with weight matrices W^l_δ and transition parameters $\mathbf{t}^l$. We define $V^l \in \mathbb{R}^{n_l \times T}$ for $l = 0 \ldots L-1$ to be a matrix of features at layer l. In case $l = 0$ (the input), the features are assumed to be real values that are defined by the user in a task-specific way. For the hidden layers ($l = 1 \ldots L-1$), we specify $V^l = \mathbb{E}\left[H^l\right]$, where $\Pr(H^l|V^{l-1})$ is defined as in Eq. (3), using the activation matrix A^l of the l^{th} layer as defined in Eq. (1).

The output $\{y_1 \ldots y_T\}$ is assumed to be a sequence of integer labels, with $y_i \in \{1 \ldots n_L\}$ written as a matrix $Y \in \mathbb{R}^{n_L \times T}$ where $Y_{it} = 1$ if $y_t = i$, and 0 otherwise. We have weight matrices W^L_δ just as with the hidden layers, and the activation matrix A^L is formed applying Eq. (1) to the features V^{L-1} of the deepest hidden layer. However now instead of a vector $\mathbf{t}$ of transition parameters, we have a full matrix $U \in \mathbb{R}^{n_L \times n_L}$. The distribution is written just as Eq. (3):

$$\Pr(Y|V^{L-1}) \propto \exp\left(\langle Y, A^L\rangle + \sum_{t=1}^{T-1}\langle Y_{*t}, U^L Y_{*(t+1)}\rangle\right)$$

but there are two important differences. First, U^L is not constrained to be diagonal, as U^l is for $l < L$. Second, while H is an arbitrary binary matrix, Y is a 0-1 matrix with a single 1 in each column, so the set of structures that are summed over for normalization is different. Instead of a set of independent binary Markov sequences, $\Pr(Y|V^{L-1})$ defines a single Markov sequence over multinomials with n_L values, so we can still efficiently compute the maximizing assignment and the probability of the correct labels with standard algorithms. The SDBN structure is illustrated in figure 1(b).

The temporal edges at internal layers of an SDBN can potentially offer distinct advantages in modeling capacity. Consider, for example, a CRF that utilizes features with a fixed temporal span over the input. The only hope to recognize patterns that occur over larger spans is via the temporal integration at the output CRF layer. A SDBN, by contrast, has the ability for its hidden units to indicate the presence of an arbitrarily long temporal pattern, or even properties of the entire sequence, owing to the earlier layers' Baum-Welch stages that can pass information over an arbitrary long temporal extent.



(a) $\delta_{\max} = 1$

(b) $\delta_{\max} = 2$

(c) $\delta_{\max} = 4$

Fig. 2. Average development set PER of the SDBN and baseline model over a range of number of layers, n_h and $\delta_{\max}$.

To train the SDBN, we first pretrain each SRBM layer with CD, using the expected values of the hidden units at each layer as the input to the next layer. Then we fine-tune all parameters by approximately maximizing $\ell = \log \Pr(\hat{Y}|V^{L-1})$ with stochastic gradient descent, where the gradient is computed using a procedure similar to error backpropagation (BP). We omit the derivation of the error gradient for lack of space, but we note that it can be computed exactly via dynamic programming in time linear in the size of the network, and is well

suited for optimized implementation via fast matrix multiplication routines and/or the use of GPU processors.

4. EXPERIMENTS

We tested the SDBN on the TIMIT phone recognition dataset.[3] The input features were 12$^{\text{th}}$ order MFCCs and energy over 25 ms windows, plus the first-order temporal differences, normalized to have zero mean and unit variance on the training data. The outputs are sequences of the standard 39-phone set of [9]. In order to model repeated phones and to get some of the modeling power of subphones, we divide each phone into two states, and constrain the model to require traversal through each substate of each phone. The boundaries between subphones are kept latent during training. We compared the complete SDBN to a baseline model that uses a sequence classifier only at the top level, exactly as if $\mathbf{t}^l$ were constrained to be zero for $l < L$.

We compare the models over a range of model depths, half-widths of the input window $\delta^1_{\max}$, and numbers of hidden units per frame (in each hidden layer). All hidden layers for $l > 1$ use $\delta^l_{\max} = 1$ Each configuration was repeated five times with different random seeds. ℓ_2 weight decay (applied after each update and scaled proportionally with T) is used for regularization and to prevent saturation of the hidden units during training. Each stage of training (that is, pretraining each layer with CD and also joint training of all parameters with BP) continued until the training criterion (squared reconstruction error for CD, log-likelihood for BP) failed to improve over five epochs, at which point the learning rate was annealed linearly to zero over another five epochs. The initial learning rates, weight decays and momentum parameters were estimated on a randomly selected 10% of the training set, which was added back before training the final model for test results.

The results are summarized in figure 2. It is apparent that using full sequence information at all layers is beneficial across nearly all configurations, and the gains are more significant as the number of hidden layers increases. The results also indicate that the use of an SDBN renders moot the need for a very wide input window: our best results are obtained with $\delta^1_{\max} = 1$, whereas $\delta^1_{\max} = 5$ or 7 is more common in related work ([8] [10]).

Our best performing configuration (150 units/frame, 8 layers, $\delta_{\max} = 1$), evaluated on the test set, achieved a PER of 25.2, which surpasses many recent systems that are highly tailored to the phone-recognition task ([11] [12] [13] [14] [15] [4] [16]) and approaches the error of the very best systems. As far as we are aware, the current state-of-the-art is the exceptional mcRBM model of [10], with a PER of 20.5. The primary innovation of that work is the use of the mean-covariance RBM in the input layer, although several other advanced techniques are used, including forced alignment of true (three-state) subphones to an HMM baseline model, and more intensive preprocessing of the input features than we have used. It also uses far more hidden units per time-frame (2048), entailing a number of parameters that is several orders of magnitude greater and therefore much slower training. In future work it would be interesting to measure the impact of temporal connections in the context of all of the advanced features present in that work.

Acknowledgments: This research was supported by NSF grant IIS-0905341. The opinions expressed in this work are those of the authors and do not necessarily reflect the views of the funding agency.

[3] We use the standard train/test split for phone recognition experiments: removing all SA records (identical sentences spoken by different speakers) from training, and testing on the core test set of 24 speakers.

5. REFERENCES

[1] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, pp. 282–289.

[2] H. Hermansky and S. Sharma, "Temporal patterns (TRAPS) in ASR of noisy speech," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2002, vol. 1, pp. 289–292.

[3] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden conditional random fields for phone classification," in *Interspeech*, 2005.

[4] Y.-H. Sung and D. Jurafsky, "Hidden conditional random fields for phone recognition," in *Automatic Speech Recognition and Understanding*, 2009.

[5] D. Yu, S. Wang, and L. Deng, "Sequential labeling using deep-structured conditional random fields," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, 2010.

[6] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, 2006.

[7] A. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *Advances in Neural Information Processing Systems 22 (Workshops)*, 2009.

[8] A. Mohamed, D. Yu, and L. Deng, "Investigation of full-sequence training of deep belief networks," in *Interspeech*, 2010.

[9] K.F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, 1989.

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

[11] J. Keshet, S. Shalev-Shwartz, S. Bengio, Y. Singer, and D. Chazan, "Discriminative kernel-based phoneme sequence recognition," in *Interspeech*, 2006.

[12] K. Crammer, "Efficient online learning with individual learning-rates for phoneme sequence recognition," *Journal of Machine Learning Research*, vol. 7, 2006.

[13] C.-C. Cheng, F. Sha, and L. K. Saul, "A fast online algorithm for large margin training of continuous-density hidden markov models," in *Interspeech*, 2009.

[14] J. Morris and E. Fosler-Lussier, "Conditional random fields for integrating local discriminative classifiers," *IEEE Trans. on Acoustics, Speech, and Language Processing*, vol. 16, no. 3, 2008.

[15] J. Keshet, D. McAllester, and T. Hazan, "Pac-bayesian approach for minimization of phoneme error rate," in *International Conference on Acoustics Speech and Signal Processing*, 2011.

[16] F. Sha and L. K. Saul, "Comparison of large margin training to other discriminative methods for phonetic recognition by hidden markov models," in *International Conference on Acoustics Speech and Signal Processing*, 2007.