Deep Segmental Neural Networks for Speech Recognition

Ossama Abdel-Hamid[1], *Li Deng*[2], *Dong Yu*[2], *Hui Jiang*[1]

[1]Department of Computer Science and Engineering, York University, Toronto, Ontario, Canada
[2] Microsoft Research, Redmond, WA USA

ossama@cse.yorku.ca, {deng, dongyu}@microsoft.com, hj@cse.yorku.ca

Abstract

Hybrid systems which integrate the deep neural network (DNN) and hidden Markov model (HMM) have recently achieved remarkable performance in many large vocabulary speech recognition tasks. These systems, however, remain to rely on the HMM and assume the acoustic scores for the (windowed) frames are independent given the state, suffering from the same difficulty as in the previous GMM-HMM systems. In this paper, we propose the deep segmental neural network (DSNN), a segmental model that uses DNNs to estimate the acoustic scores of phonemic or sub-phonemic segments with variable lengths. This allows the DSNN to represent each segment as a single unit, in which frames are made dependent on each other. We describe the architecture of the DSNN, as well as its learning and decoding algorithms. Our evaluation experiments demonstrate that the DSNN can outperform the DNN/HMM hybrid systems and two existing segmental models including the segmental conditional random field and the shallow segmental neural network.

Index Terms: Segmental Model, Segmental Conditional Random Field, Deep Segmental Neural Network

1. Introduction

Recently, deep-neural-network hidden Markov model (DNN/HMM) hybrid systems have achieved remarkable performance in many large vocabulary speech recognition tasks [1, 2, 3, 4, 5, 6, 7, 8]. These DNN/HMM hybrid systems, however, estimate the observation likelihood score for each (windowed) frame independently, and rely on a separate HMM to connect these scores to form the overall scores for phonemes, words, and then sentences.

It has been known for decades that modeling speech using the conventional HMM has several limitations as analyzed in [9, 10, 11]. The limitations include the assumption of conditional independence of temporal observations given the state, the restriction of using frame-level features, and weak duration modeling. To eliminate these limitations, many techniques have been developed. These techniques can be described in a unified framework named the segmental model [9]. The state sequence in the segmental models is often modeled as a Markov chain. However, these states emit variable-length segments (typically phonemes or subphonemes) instead of a set of independent frames. Because of this characteristic, segment-level features such as duration can be easily incorporated in the segmental models and the frame independence assumption is no longer needed.

More recently, segmental models have also been developed in the discriminative model framework (e.g., segmental conditional random field (SCRF) [12, 13]). These models, however, are typically shallow, require manual feature design, and are often used in the second pass decoding scenario. In these models, the feature design and the log-linear classifier are independently trained as two separate components of the system.

In this paper, we propose an integrated segmental model — deep segmental neural network (DSNN). Similar to the SCRF, at the top of the DSNN is a conditional random field (CRF) that models sequences. Unlike the SCRF, our proposed DSNN uses a DNN to model the variable-length segments and learn the CRF and DNN parameters jointly. Compared to the DNN/HMM hybrid system, the DSNN replaces the HMM with a CRF and generates a score for each variable-length segment instead of for each frame. These acoustic scores, one for each segment, are combined with the language model (LM) scores to compute the label sequence's conditional probability.

The rest of the paper is organized as follows. In Section 2 we describe the proposed DSNN in detail. We also propose three simple ways to reduce the model complexity. In Sections 3 and 4, we introduce the learning and decoding algorithms we have developed for the DSNN. We report experimental results on the TIMIT dataset in Section 5 and demonstrate that the DSNN performs better than the DNN/HMM hybrid systems and the SCRF. We discuss the related work in Section 6 and conclude the paper in Section 7.

2. The deep segmental neural network

2.1. Model description

Assuming we are given a sequence of feature vectors, X, for an utterance, we use $L = \{l_1, \cdots, l_K\}$ to represent a sequence of labels, which may be defined at the subphoneme, phoneme, syllable or even word level, and $T = \{t_0, t_1, \cdots, t_K\}$ to denote one particular time alignment for the label sequence. The label sequence and the associated time sequence form a segment sequence. The conditional probability for the segment sequence Y given the speech utterance X is estimated as

$$P(L,T|X) = \frac{\exp\left(\sum_i s(l_i, t_{i-1}+1, t_i|X) + u(L)\right)}{\sum_{\hat{L},\hat{T}} \exp\left(\sum_j s(\hat{l}_i, \hat{t}_{i-1}+1, \hat{t}_i|X) + u(\hat{L})\right)}, \tag{1}$$

where $s(l_i, t_{i-1}+1, t_i|X)$ represents the acoustic score of getting label l_i for the segment that has the time boundaries $[t_{i-1}+1, t_i]$, and $u(L)$ stands for the total LM score computed for the entire label sequence L. The denominator in Eq.(1) sums over all possible label sequences $\hat{L}$ and time alignments $\hat{T}$. If we are only interested in the label sequence L, we can sum over all possible time alignments to yield the posterior probability of

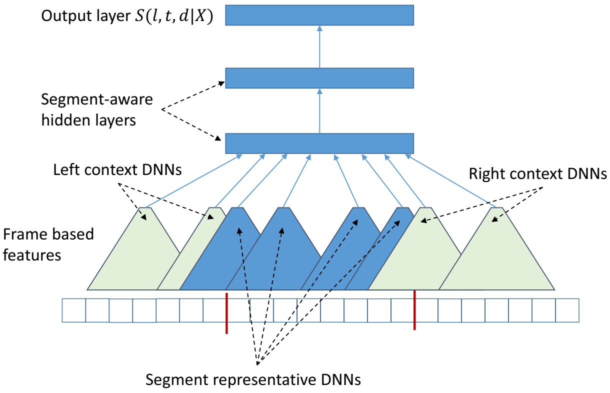

Output layer $S(l,t,d|X)$
Segment-aware hidden layers
Left context DNNs
Right context DNNs
Frame based features
Segment representative DNNs

Figure 1: *Structure of the deep segmental neural network (DSNN)*

one particular L given X as

$$\begin{aligned} P(L|X) &= \sum_T P(L,T|X) \\ &= \frac{\sum_T \exp\left(\sum_i s(l_i, t_{i-1}+1, t_i|X) + u(L)\right)}{\sum_{\hat{L},\hat{T}} \exp\left(\sum_j s(\hat{l}_i, \hat{t}_{i-1}+1, \hat{t}_i|X) + u(\hat{L})\right)}. \end{aligned} \quad (2)$$

In this work, we use DNNs to compute acoustic scores for each variable-length segment, $s(l_i, t_{i-1}+1, t_i|X)$, and thus name our model the *deep segmental neural network (DSNN)*. The scores used here may take values of any suitable range and they need not be log probabilities. The total acoustic and language score of a label and segmentation sequence is the negated value of the energy function of the model. We leave the DNN to compute any scores that maximize conditional probability of the training data. Note that any type of LM can be used in the above definition. In this paper, we use a simple phoneme bigram LM to compute $u(L)$. Other more complex LMs can be used as well but they may require some approximations such as constraining search space with word graphs instead of summing over all possible segment sequences.

2.2. Score functions

It is well known that speech segments are of variable length. However, the DNN expects fixed-length inputs. This imposes a challenge when we apply DNNs to segmental models. In this section, we propose methods to normalize segments, which lead to various practical ways of implementing the DSNN.

The basic structure of the DSNN, shown in Fig. 1, is used to compute the acoustic scores for segments. Some DNNs, represented as the trapezoid shapes in the figure, are used to compute frame-level features. Each of these DNNs includes a few fully-connected hidden layers. Similar to the DNN/HMM hybrid system, each DNN computes an acoustic score by taking several consecutive frames within a context window, which is centered at one particular frame located within the given segment or the left/right context of the segment. To normalize variable-length segments, we use N_c DNNs distributed evenly over the segment frames as shown in Fig. 2. Moreover, we add N_l DNNs to process the left context and N_r DNNs to process the right context.

Example values used in this work are $N_l = N_r = 2$ and $N_c = 4$. The outputs from these DNNs are then fed into one or more layers of additional hidden nodes, which now take a fixed-size input. As shown in Fig. 1, these upper layers are called segment-aware hidden layers, on top of which the output layer is added to compute the final label score vector corresponding to the current segment.

The weights of the lower-level DNNs may be tied. In this case, a single DNN is shifted along the time axis in the speech utterance to compute a fixed-size feature as the input of the upper-level, segment-aware hidden layers in the DSNN.

This segment aware-DNN requires re-computation of the DNN outputs for every segment boundary. Otherwise, a simpler method is to use a frame-based DNN to compute acoustic scores for every frame. To compute the score $o(l,t)$ of label l at time t, the DNN takes a number of consecutive frames centred at time t. Then a segment score is derived from these frame-based scores using several alternative methods as illustrated in Fig. 2 and described below.

2.2.1. *Approximation by the score from the middle frame*

The first method, shown in Fig. 2a, approximates the segment's score using the DNN score computed for the middle frame within the segment; i.e.,

$$s(l_i, t_{i-1}+1, t_i|X) = o(l_i, \frac{t_i + t_{i-1} + 1}{2}) \quad (3)$$

2.2.2. *Approximation by the score from the final frame*

The segment's score can also be approximated by the DNN score computed from the final frame of the segment as shown in Fig. 2b as

$$s(l_i, t_{i-1}+1, t_i|X) = o(l_i, t_i) \quad (4)$$

2.2.3. *Approximation by summing scores from full segment*

Similarly, the segment's score can be approximated by summing the DNN scores over all frames located within the segment as shown in Fig. 2c

$$s(l_i, t_{i-1}+1, t_i|X) = \sum_{t=t_{i-1}+1}^{t_i} o(l_i, t_i) \quad (5)$$

3. Training of weights via backpropagation

In this section, we describe the learning method for estimating the weights in the DSNN model from training data. For each utterance in the training set, we have its feature sequence, X, and label sequence, L. No segment's time boundary information, T, is given during training. The DSNN weights are learned discriminatively to maximize the label sequences' conditional likelihood function in Eq. (2). This objective function is optimized in this work using the stochastic gradient ascent method.

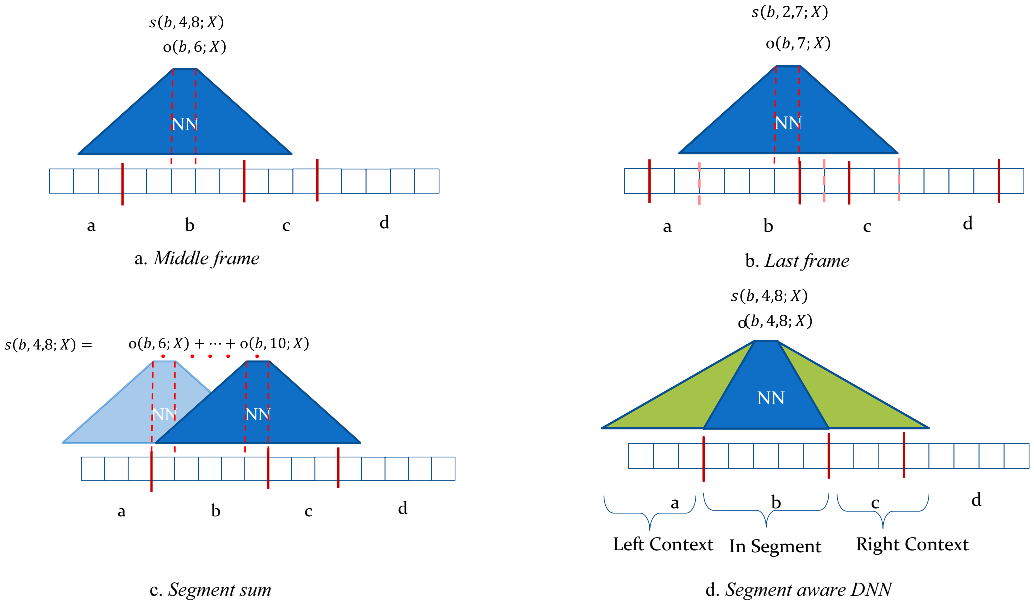
For any particular weight matrix, $\mathbf{W}$, in the DSNN, the derivative of logarithm of the objective function can be computed based on the chain rule as follows:

$$\frac{\partial \log p(L|X)}{\partial \mathbf{W}} = \sum_{l,t_s,t_e} \frac{\partial \log p(L|X)}{\partial s(l,t_s,t_e)} \cdot \frac{\partial s(l,t_s,t_e)}{\partial \mathbf{W}} \quad (6)$$

where $s(l,t_s,t_e)$ denotes the segmental acoustic score computed by the low-level DNN defined by $\mathbf{W}$.

The first derivative in the right hand-side of Eq.(6) can be computed based on Eq.(2) as follows:

$$\frac{\partial \log p(L|X)}{\partial s(l,t_s,t_e)} = \frac{\sum_{T\in A} p(L,T|X)}{p(L|X)} - \sum_{(\hat{L},\hat{T})\in B} p(\hat{L},\hat{T}|X) \quad (7)$$



a. *Middle frame*

b. *Last frame*

c. *Segment sum*

d. *Segment aware DNN*

Figure 2: *Three different methods for approximating the score of a segment (a, b, c). The corresponding segment-aware NN in d.*

where A denotes the set of time alignments that assign time boundaries $[t_s, t_e]$ with label l, and B denotes the set of all possible label segments and time alignments that embed $(l, [t_s, t_e])$). The summations in Eq. (7) contain an exponentially increasing number of terms. However, if a bigram language model is used in Eq. (2), these summations can be recursively evaluated using the forward-backward algorithm.

In this case, we define $\alpha_s(l, t)$ as the sum of partial scores of all paths that lead to label l starting at time t excluding the current label score. We also define $\alpha_e(l, t)$ as the sum of partial scores of all paths that end with a segment label l and end at time instant t. Figure 3 illustrates one step in computing $\alpha_s(l, t)$, which accounts for all labels before time t, and one step in computing $\alpha_e(l, t)$, which considers all different lengths of segment l ending in time t. These two quantities can be computed recursively according to

$$\alpha_s(l,t) = \sum_{\hat{l}} \alpha_e(\hat{l}, t-1) \exp\left(w(l;\hat{l})\right) \qquad (8)$$

and

$$\alpha_e(l,t) = \sum_{d=1}^{D_l} \alpha_s(l, t-d+1) \exp\left(s(l, t-d+1, t|x)\right) \quad (9)$$

where d represents the segment duration, D_l is the maximum duration for label l (which can be learned from the training data, and $w(l;\hat{l})$ is the language model score for transitioning from label $\hat{l}$ to l.

Similarly, β_s and β_e are defined for the backward direction as:

$$\beta_e(l,t) = \sum_{\hat{l}} \beta_s(\hat{l}, t+1) \exp\left(w(\hat{l};l)\right) \qquad (10)$$

$$\beta_s(l,t) = \sum_{d=1}^{D_l} \beta_e(l, t+d-1) \exp\left(s(l, t, t+d-1|x)\right) \quad (11)$$

Model learning requires the computation of $s(l, t_s, t_e)$ for all possible l, t_s, and t_e, where the duration of each label l is limited to the maximum duration seen for the label in the training set. This computation has been efficiently implemented by parallelizing them in a GPU. After these computations, the derivatives of the log objective function are back-propagated to all DNNs to update each weight matrix via gradient ascent.

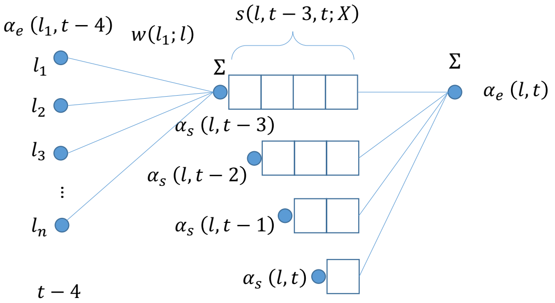


Figure 3: *Illustration of recursive "forward" computations of α_s and α_e.*

4. Decoding

In decoding, we aim to search for the best label and alignment sequence for each speech utterance X in the test set. With the use of a bigram language model, the search can be carried out using the Viterbi version of the forward algorithm in Eqs. (8) and (9) by replacing summation with maximization. This decoding is much slower than the standard HMM Viterbi algorithm as it requires the consideration of all possible segment durations. In our experiments, we have speeded up decoding considerably using parallel codes on both CPU (for Viterbi search) and GPU (for computing DSNN segments' scores).

5. Experimental evaluation

5.1. Experimental setup

Experiments are performed on the TIMIT corpus in the standard phone recognition task with the core test set and with 39 folded classes. In feature extraction, speech is analyzed using a 25-ms

Table 1: *Phone error rate (PER) comparisons of the full version of a DSNN and several approximate, simplified versions.*

Score Function	LM	no LM
Hybrid DNN-HMM	23.31%	24.63%
Simplified DSNN - Middle Frame	25.61%	24.72%
Simplified DSNN - Last Frame	24.59%	25.36%
Simplified DSNN - Segment Sum	25.42%	25.35%
Full-scale DSNN	22.90%	23.92%

Hamming window with a 10-ms fixed frame rate. The speech feature vector is generated by a Fourier-transform-based filterbanks, which includes 40 coefficients distributed on a Mel scale and energy, along with their first and second temporal derivatives.

In the DSNN experiments, only label sequences are used for training and no alignment information is used. Furthermore, a phoneme bigram language model estimated from the training set is used to compute the label sequence's log probability in Eq. 1 as the LM score. This LM score is used in both training and decoding. No duration model is used for any model. During DSNN training, a learning rate annealing and early stopping strategy are adopted following [14].

5.2. Results

Experiments are conducted to measure the performance of the proposed DSNN and to evaluate different approximated score functions and DNN architectures. Table 1 summarizes the results and compares the DSNN to the hybrid DNN/HMM model. All the models used in the experiments have 4 fully connected hidden layers. We observe that the full-scale DSNN (with a segment-aware DNN) outperforms the hybrid DNN/HMM model and all approximate versions with various simplified segment score estimation methods. Unfortunately, none of the simplified approach explored here can beat the hybrid DNN/HMM model.

Table 2 shows the performance of the full-scale DSNN with different architectures and hyper-parameters. Use of four hidden layers performs considerably better than two. Moreover, use of different sets of weights for each of the low-level, frame-based DNNs ("non-shared" in column 2) performs better than sharing weights (row 4 vs. row 3), and also reduces the complexity in computing the DSNN scores. While a lower PER of 21.87% was obtained using a convolutional neural network [15], the DSNN (with no convolutional structure) performs significantly better than other segmental models such as the Segmental CRF (PER 33.1%) [16] and the shallow segmental neural network (SNN) (performs significantly worse than the hybrid system) [17].

6. Relation to prior work

While both using a segmental structure, the DSNN described in this paper is different from the earlier model of SCRF [16, 18] in several ways. First, feature transformation and the sequence model component in the DSNN are optimized jointly, while in the SCRF they are two separate processes and the features are often manually defined. Second, we used a conditional likelihood function (Eq. 1) that allows for including an arbitrary LM. In contrast, the LM for a SCRF is defined using the transitional features and their weights between two states. Although by carefully designing the model states, this can map to N-gram LMs [12] in an indirect way, we believe that our segmental model formulation of Eq. 1 is more natural for incorporating arbitrary LMs (e.g., Recurrent neural net LMs).

Table 2: *PER comparisons among different DSNN architectures. The first column shows the number of hidden units in each hidden layer. The two pairs of brackets represent the lower DNN and the top segment-dependant neural net, respectively.*

DSNN Architecture	features sharing	PER
{300}, {1000}	shared	24.15%
{300*8}, {1000}	non-shared	24.40%
{1000,500}, {1000,1000}	shared	23.52%
{1000,150*8}, {1000,1000}	non-shared	22.90%
{CNN (84 Kernels * 20 bands),150*8}, {1000,1000}	non-shared	21.87%

A similar deep model to ours has been proposed for the CRF model in [19, 20, 21, 22] with the difference of being frame-based rather than being segment-based. Separately, in [17], a segmental neural net model was proposed where the variable length segment was sampled to a fixed number of frames and where some frames may be skipped or repeated. In the DSNN presented in this paper, we do re-sampling on the hidden layer features that represent a sequence of frames. So, theoretically all frames can be represented in the DSNN while preserving the structure between consecutive frames.

7. Conclusions

We have presented a novel segmental model — the deep segmental neural network. The DSNN estimates the acoustic scores for variable-length segments and models the label sequence's conditional probability directly. This eliminates the assumption that each frame is independent of each other given the state and thus has potential to perform better than the DNN/HMM hybrid. We have described several possible simplifications for the segment score estimation and the DSNN learning and decoding algorithms. We demonstrated that the unsimplified DSNN performs better than the hybrid DNN/HMM on the TIMIT phone recognition task. While this is an initial attempt to use the DSNN, the results are promising and are better than that obtained by other segmental models such as the SCRF [16] and the SNN [17].

The proposed DSNN can be further improved in several directions. First, the DSNN without using convolution layers performs worse than the CNN. A natural extension is to use CNN in the DSNN architecture. Second, the approximated score functions explored in this work do not work well. Additional work needs to be done to find a better simplified method for segment score calculation. Third, we only used acoustic and language model information in this work. Additional improvement may come from exploiting other information such as duration.

8. Acknowledgments

We would like to thank Drs. Alex Acero and Geoffrey Zweig at Microsoft Research for valuable discussion and suggestions.

9. References

[1] G. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30 –42, jan. 2012.

[2] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using context-dependent deep neural networks,” in *Interspeech 2011*, 2011.

[3] 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.

[4] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82 –97, nov. 2012.

[5] D. Yu, L. Deng, and F. Seide, “The deep tensor neural network with applications to large vocabulary speech recognition,” *IEEE Trans. Audio, Speech, and Language Proc.*, vol. 21, no. 2, pp. 388–396, 2013.

[6] L. Deng, O. Abdel-Hamid, and D. Yu, “A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013.

[7] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, , and A. Acero, “Recent advances in deep learning for speech research at Microsoft,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013.

[8] S. M. Siniscalchi, D. Yu, L. Deng, and C. hui Lee, “Speech recognition using long-span temporal patterns in a deep network model,” *IEEE Signal Processing Letters*, March 2013.

[9] M. Ostendorf, V. Digalakis, and O. Kimball, “From HMM’s to segment models: A unified view of stochastic modeling for speech recognition,” *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 5, pp. 360–378, 1996.

[10] L. Deng, “A generalized hidden markov model with state-conditioned trend functions of time for the speech signal,” *Signal Processing*, vol. 27, no. 1, pp. 65 – 78, 1992.

[11] L. Deng, M. Aksmanovic, X. Sun, and C. Wu, “Speech recognition using hidden markov models with polynomial regression functions as nonstationary states,” *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 4, pp. 507 –520, oct 1994.

[12] G. Zweig and P. Nguyen, “A segmental CRF approach to large vocabulary continuous speech recognition,” in *Automatic Speech Recognition Understanding, 2009. ASRU 2009. IEEE Workshop on*, 13 2009-dec. 17 2009, pp. 152 –157.

[13] G. Zweig, P. Nguyen, D. Van Compernolle, K. Demuynck, L. Atlas, P. Clark, G. Sell, M. Wang, F. Sha, H. Hermansky, D. Karakos, A. Jansen, S. Thomas, G. Sivaram, S. Bowman, and J. Kao, “Speech recognition with segmental conditional random fields: A summary of the JHU CLSP 2010 summer workshop,” in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, may 2011, pp. 5044 –5047.

[14] A. Mohamed, G. Dahl, and G. Hinton, “Acoustic modeling using deep belief networks,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14 –22, jan. 2012.

[15] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, “Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, march 2012, pp. 4277 – 4280.

[16] G. Zweig, “Classification and recognition with direct segment models,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, march 2012, pp. 4161 –4164.

[17] S. Austin, G. Zavaliagkos, J. Makhoul, and R. Schwartz, “Continuous speech recognition using segmental neural nets,” in *Neural Networks, 1992. IJCNN., International Joint Conference on*, vol. 2, jun 1992, pp. 314 –319 vol.2.

[18] Y. He and E. Fosler-Lussier, “Efficient segmental conditional random fields for one-pass phone recognition,” in *Interspeech 2012*, 2012.

[19] R. Prabhavalkar and E. Fosler-Lussier, “Backpropagation training for multilayer conditional random field based phone recognition,” in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, march 2010, pp. 5534 –5537.

[20] A.-R. Mohamed, D. Yu, and L. Deng, “Investigation of full-sequence training of deep belief networks for speech recognition,” in *Interspeech*, 2010, pp. 2846–2849.

[21] D. Yu and L. Deng, “Deep-structured hidden conditional random fields for phonetic recognition,” in *Interspeech*, 2010.

[22] T.-M.-T. Do and T. Artieres, “Neural conditional random fields,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, vol. 9, 5 2010.