Investigation of Full-Sequence Training of Deep Belief Networks for Speech Recognition

Abdel-rahman Mohamed 1*, Dong Yu 2, Li Deng 2

1 Department of Computer Science, University of Toronto, Toronto, ON Canada
2 Microsoft Research, Redmond, WA USA

asamir@cs.toronto.edu, {dongyu,deng}@microsoft.com

Abstract

Recently, Deep Belief Networks (DBNs) have been proposed for phone recognition and were found to achieve highly competitive performance. In the original DBNs, only frame-level information was used for training DBN weights while it has been known for long that sequential or full-sequence information can be helpful in improving speech recognition accuracy. In this paper we investigate approaches to optimizing the DBN weights, state-to-state transition parameters, and language model scores using the sequential discriminative training criterion. We describe and analyze the proposed training algorithm and strategy, and discuss practical issues and how they affect the final results. We show that the DBNs learned using the sequence-based training criterion outperform those with frame-based criterion using both three-layer and six-layer models, but the optimization procedure for the deeper DBN is more difficult for the former criterion.

Index Terms: Deep Belief Networks, phone recognition, discriminative training, full-sequence optimization

1. Introduction

Research in speech recognition has explored layered architectures in the recognizer design for quite some time (e.g., [1][2][3][5][15][17][18][19][20]), motivated partly by the desire to capitalize on some analogous properties in the human speech generation and perception systems. In these studies, learning of model parameters has been one of the most prominent and difficult problems and has thus received much attention. In parallel with the development in speech recognition research, a recent progress in learning methods from neural network research has renewed the interest in further exploring deep structured models. One major advance is the development of highly effective learning techniques for the deep belief networks (DBNs), which are densely connected, directed belief nets with many hidden layers [8][9].

Although deep networks typically have higher modeling power than their shallow counterparts, learning in the deep networks is significantly harder because it is difficult to infer the posterior distribution over the hidden variables when given a data vector and because the simple back-propagation algorithm does not perform effectively due to the significantly increased chance of trapping into a local optimum. The recently proposed learning strategy -- generative pre-training using restricted Boltzmann machines (RBMs) and contrastive divergence followed by discriminative fine-tuning -- is a significant advance in learning DBNs. This strategy has proved to be effective in a number of applications including coding and classification for speech, audio, and image [6][8][9][12][13][14].

In a nutshell, DBNs can be considered as a highly complex nonlinear feature extractor where each layer of the hidden units learns to represent features that capture higher order correlations in the original input data. DBNs have been recently proposed for phone recognition and were found to achieve very competitive performance [13]. In these DBNs, only frame-level information was used for training DBN weights, the transition parameters and language model (LM) scores were obtained from an HMM-like approach and were trained independently of the DBNs. However, speech recognition is a sequential or full-sequence learning problem and it has been well known that discriminative information at the sequence level contributes to improving recognition accuracy (e.g., [7]). Using neural networks to discriminate between sequences has indeed been proposed in the past (e.g., [4][10][11][16]), but it was done in shallow architectures and not in an integrative manner. In this paper, we investigate approaches to optimizing the DBN weights, transition parameters, and language model (LM) scores jointly using a discriminative training criterion at the sequence level designed specifically for DBNs.

The rest of the paper is organized as follows. In Section 2 we introduce the DBN, with the RBMs as its constituents, and the general learning strategy for the DBN. In Section 3 we describe and analyze the sequential learning algorithm we developed for the DBN and discuss the practical issues and training procedures. We present the experimental results on the TIMIT phonetic recognition task as our initial exploration in Section 4, and conclude the paper in Section 5.

2. Deep Belief Network Basics

DBNs are densely connected, directed belief nets with many hidden layers for which learning is a difficult problem. In this section we introduce an effective training algorithm that learns each layer greedily by treating each pair of layers as an RBM before doing a joint optimization of all the layers.

2.1. Restricted Boltzmann Machines

An RBM is a particular type of Markov random field (MRF) that has one layer of (typically Bernoulli) stochastic hidden units and one layer of (typically Bernoulli or Gaussian) stochastic visible units. RBMs can be represented as bipartite graphs since all visible units are connected to all hidden units, but there are no visible-visible or hidden-hidden connections. In the RBMs, the joint distribution p(v, h; θ) over the visible units v and hidden units h, given the model parameters θ, is defined in terms of an energy function E(v, h; θ) of

\[ p(v, h; \theta) = \frac{\exp(-E(v, h; \theta))}{Z} \]  

(1)
where \( Z = \sum_{a} \sum_{h} \exp \left( -E(v, h; \theta) \right) \) is a normalization factor or partition function, and the marginal probability that the model assigns to a visible vector \( v \) is
\[
p(v; \theta) = \frac{\sum_{h} \exp \left( -E(v, h; \theta) \right)}{Z}.
\]

For a Bernoulli (visible)-Bernoulli (hidden) RBM, the energy is
\[
E(v, h; \theta) = -\sum_{i=1}^{V} \sum_{j=1}^{H} w_{ij} v_i h_j - \sum_{i=1}^{V} b_i v_i - \sum_{j=1}^{H} a_j h_j.
\]

where \( w_{ij} \) represents the symmetric interaction term between visible unit \( v_i \) and hidden unit \( h_j \), \( b_i \) and \( a_j \) the bias terms, and \( V \) and \( H \) are the numbers of visible and hidden units. The conditional probabilities can be efficiently calculated as
\[
p(h_j = 1 | v; \theta) = \sigma \left( \sum_{i=1}^{V} w_{ij} v_i + a_j \right),
\]
\[
p(v_i = 1 | h; \theta) = \sigma \left( \sum_{j=1}^{H} w_{ij} h_j + b_i \right),
\]
where \( \sigma(x) = 1 / (1 + \exp(x)) \).

Similarly, for a Gaussian-Bernoulli RBM, the energy is
\[
E(v, h; \theta) = -\sum_{i=1}^{V} \sum_{j=1}^{H} w_{ij} v_i h_j + \frac{1}{2} \sum_{i=1}^{V} (v_i - b_i)^2 - \sum_{j=1}^{H} a_j h_j.
\]

The corresponding conditional probabilities become
\[
p(h_j = 1 | v; \theta) = \mathcal{N} \left( \sum_{i=1}^{V} w_{ij} v_i + a_j, 1 \right),
\]
\[
p(v_i = 1 | h; \theta) = \mathcal{N} \left( \sum_{j=1}^{H} w_{ij} h_j + b_i, 1 \right),
\]
where \( v_i \) takes real values and follows a Gaussian distribution with mean \( \sum_{j=1}^{H} w_{ij} h_j + b_i \) and variance one. Gaussian-Bernoulli RBMs can be used to convert real-valued stochastic variables to binary stochastic variables which can then be further processed using the Bernoulli-Bernoulli RBMs.

Following the gradient of the log likelihood \( \log p(v; \theta) \) we obtain the update rule for the weights as [8]:
\[
\Delta w_{ij} = (v|h)_{data} - (v|h)_{model},
\]
where \( (v|h)_{data} \) is the expectation observed in the training set and \( (v|h)_{model} \) is the same expectation under the distribution defined by the model. Unfortunately, \( (v|h)_{model} \) is extremely expensive to compute exactly so the contrastive divergence (CD) approximation to the gradient is used where \( (v|h)_{model} \) is replaced by running the Gibbs sampler initialized at the data for one full step [9].

### 2.2. Deep Belief Networks

From the decoding point of view, a DBN can be treated as a multi-layer percoptron with many layers. The input signal is processed layer by layer with Eq. (4) till the final layer, whose output is then transformed into a multinomial distribution using the softmax operation
\[
p(l = k | h; \theta) = \frac{\exp \left( \sum_{i=1}^{M} \lambda_{ki} h_i \right)}{Z(h)},
\]
where \( l = k \) denotes the input been classified into the \( k \)-th class, and \( \lambda_{ki} \) is the weight between hidden unit \( h_i \) at the last layer and class label \( k \).

As the baseline system, we use the conventional frame-level criterion to train the DBNs. Specifically, we adopt the strategy proposed in [13][8][9] for training DBN weights. That is, we first train a stack of RBMs in a generative manner and then fine-tune all the parameters jointly using back-propagation algorithm by maximizing the frame-level cross-entropy between the true and the predicted probability distributions over class labels.

### 3. Full-Sequence Training of DBNs

The standard discriminative back-propagation step as used in [8][9][13] optimizes the log posterior probability \( p(l_t \mid v_t) \) of class labels given the current input, both at time-frame \( t \) (which may be a fixed local block of frames). We call this method of training DBNs a frame-based approach, because it uses the frame (or frame-block) only to predict the class labels. It does not explicitly use the fact that the neighboring frames (or frame-blocks) have smaller distances between the assigned probability distributions over class labels. To take this fact into account we model the probability of the whole sequence of labels given the whole utterance \( p(l_1:T \mid v_{1:T}) \).

The approach we take in this paper is to consider the top-most layer of the DBN as a linear-chain conditional random field (CRF) with \( h_t \) as input features from the lower layer at time \( t \). This new model can be seen as a modification of the deep-structured CRF of [19][20] where the lower multiple layers of CRFs are replaced by DBNs. The resulting architecture is shown in Fig. 1, which can be equivalently seen as shared DBNs unfolding over time.

The conditional probability of the full-sequence labels given the full-sequence input features in this sequential model is
\[
p(l_1:T \mid v_{1:T}) = \frac{\exp \left( \sum_{t=1}^{T} \gamma_{l_t} \phi_{l_t}(l_{t-1}, l_t) + \sum_{t=1}^{T} \lambda_{d_t} \phi_{d_t}(h_t) \right)}{Z(l_{1:T})},
\]
where the transition feature is
\[
\phi_{l_t}(l_{t-1}, l_t) = \begin{cases} 1 & \text{if } l_{t-1} = l \text{ and } l_t = j \\ 0 & \text{otherwise} \end{cases},
\]
\( \gamma_{l_t} \) is the parameter associated with this transition feature, \( \phi_{d_t} \) is the dimension of the hidden unit value at the \( t \)-th frame at the last layer \( h_t \). \( D \) is the dimension of (or number of units) at that hidden layer.

![Fig. 1: Architecture of the DBN developed and evaluated in this work.](image-url)
### 4.1. Experimental Setup

Phone recognition experiments were performed on the TIMIT corpus. The standard training set consisting of 462 speakers was used for training DBNs, with all SA removed. The standard development set of 50 speakers was used for model tuning. Results are reported using the standard 24-speaker core test set consisting of 192 sentences with 7,333 phone tokens. The speech was analyzed using a 25-ms Hamming window with a 10-ms fixed frame rate. In all the experiments, we represented the speech using first- to 12th-order Mel frequency cepstral coefficients (MFCCs) and energy, along with their first and second temporal derivatives. The data were normalized to have zero mean and unit variance. All experiments used a context window of block of 11 frames as the visible states. We used 183 target class labels (i.e., three states for each of the 61 phones). After decoding, the 61 phone classes were mapped to a standard set of 39 classes for scoring using the HRes tool in HTK, in the same way as other standard TIMIT experiments (e.g., [5]).

### 4.2. DBN Training

All of our experiments used a bigram language model over phones, which was jointly estimated with all DBN parameters using the training set. The decoder parameters were tuned to optimize performance on the development set for each run using grid search. Training of the DBNs was accelerated by exploiting a graphics processor. A single pass over the entire training set during pre-training took about 3-4 minutes per layer. An epoch of fine-tuning with back-propagation took around 10 and 25 minutes with frame-based, and sequence-based criteria, respectively.

![Fig. 2: Log conditional likelihood values in fine tuning](image)

![Fig. 3: Cross-entropy values observed in fine tuning](image)

The behavior of the training process is carefully examined. As an example, we plot in Fig. 2 the training objective -- the log conditional likelihood -- as a function of fine-tuning epochs in the training, development, and test sets, respectively. The same plot for cross entropy is shown in Fig. 3.

### 4.3. Phonetic Recognition Results

Table 1 shows TIMIT phone recognition accuracy using frame-based and sequence-based training criteria with three-layer DBNs (i.e., one hidden layer). The new full-sequence training improves phone recognition accuracy for both dev and test sets, although more for the dev set. A relatively small amount of tuning is required to achieve such improvement.
Table 1. Comparative TIMIT phone recognition accuracy with three-layer DBNs

<table>
<thead>
<tr>
<th></th>
<th>Dev</th>
<th>Core Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Three-layer DBN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frame-based</td>
<td>75.60%</td>
<td>73.87%</td>
</tr>
<tr>
<td>Sequence-based</td>
<td>76.82%</td>
<td>74.10%</td>
</tr>
</tbody>
</table>

When the DBN goes deeper to six layers, the recognizer’s accuracy is sharply increased to over 77% for the frame-based system, as shown in the first row of Table 2, consistently with the finding in [13]. Without careful tuning of the training parameters, the full-sequence training was not able to improve recognition accuracy in the initial set of experiments. Careful adjustment of a set of parameters based on the dev set gives rise to higher accuracy. One of the parameters is the mini-batch size (i.e. the number of sentences in each mini batch) used in the gradient descent training of the DBN weights and integrated CRF/LM transition weights. With adjustment of the corresponding learning rate, use of increasing mini-batch sizes improves the accuracy to 77.81% on the core test set, as shown in Table 2. We also observed the importance of constraining the transition parameters. Without such a constraint, the accuracy drops to 76.22%

Further optimization of the pre-training and fine-tuning steps [13] for frame-based training is able to improve the core-set accuracy to 77.77%. A rather different kind of optimization during the pre-training stage with bi-directionally connected fully discriminative have now become largely generative described in of [19][20]. The top layer’s linear train the DBN weights gives rise to an architecture that can be discriminative learning approach to jointly optimizing the training. Specifically, we develop and present a full-sequence training to replace the randomization in the pre-training.

Table 2. Comparative TIMIT phone recognition accuracy with six-layer DBNs.

<table>
<thead>
<tr>
<th>Six-layer DBN</th>
<th>Dev</th>
<th>Core Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame-based</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB Size=1</td>
<td>78.61%</td>
<td>77.15%</td>
</tr>
<tr>
<td>MB Size=5</td>
<td>79.30%</td>
<td>77.31%</td>
</tr>
<tr>
<td>MB Size=150</td>
<td>79.01%</td>
<td>77.62%</td>
</tr>
<tr>
<td>MB Size=800</td>
<td>78.79%</td>
<td>77.81%</td>
</tr>
</tbody>
</table>

5. Conclusions

This paper presents our recent work on extending the DBN method of [13] from frame-based training to sequence-based training. Specifically, we develop and present a full-sequence discriminative learning approach to jointly optimizing the DBN weights, state-to-state transition parameters, and language model scores.

Setting up the full-sequence discriminative criterion to train the DBN weights gives rise to an architecture that can be considered as a modification of the deep-structured CRF described in of [19][20]. The top layer’s linear-chain CRF remains the same but the lower layers of deep CRFs that are fully discriminative have now become largely generative during the pre-training stage with bi-directionally connected DBNs. Not only does this change overcome the difficulty for unsupervised learning in the intermediate layers of deep CRFs, but it also allows the use of longer windows in the input data. Further, DBNs appear to apportion the power of discriminative and generative modeling in a more balanced way than deep CRF. Further improvement includes using the most likely label sequences that are consistent with the reference during optimization rather than sticking to a single alignment as the reference label sequence. We are currently also exploring other successful techniques in speech recognition in the DBN framework, such as the use of object functions better correlating with errors than MMI [7], use of context dependency [1][2], and conditioning the DBN weights on specific auxiliary factors that control the variability in the speech data [21][22].

6. References