INVESTIGATION OF DEEP BOLTZMANN MACHINES FOR PHONE RECOGNITION

Zhao You, Xiaorui Wang, Bo Xu

Interactive Digital Media Technology Research Center,
Institute of Automation, Chinese Academy of Sciences,
Beijing, P.R.China
{zhao.you, xiaorui.wang, xubo}@ia.ac.cn

ABSTRACT

In the past few years, deep neural networks (DNNs) achieved great successes in speech recognition. The layer-wise pre-trained deep belief network (DBN) is known as one of the critical factor to optimize the DNN. However, the DBN has one shortcoming that the pre-training procedure is in a greedy forward pass. The top-down influences on the inference process are ignored, thus the pre-trained DBN is suboptimal. In this paper, we attempt to apply deep Boltzmann machine (DBM) on acoustic modeling. DBM has the advantages that a top-down feedback is incorporated and the parameters of all layers can be jointly optimized. Experiments are conducted on the TIMIT phone recognition task to investigate the DBM-DNN acoustic model. Comparing with the DBN-DNN with same amount of parameters, phone error rate on the core test set is reduced by 3.8% relatively, and additional 5.1% by dropout fine-tuning.

Index Terms— phone recognition, acoustic modeling, Deep Boltzmann Machines, Deep Neural Networks

1. INTRODUCTION

In the past few years, deep neural networks (DNNs) were introduced to speech recognition tasks and achieved great successes. The DNN-HMM acoustic models achieved significant recognition error reduction over discriminatively trained GMM-HMM models [1]. It is believed that the efficient and powerful modeling ability of deeper networks is one critical factor to the remarkable accuracy gains [2, 3].

A two-stage training procedure, pre-training and fine-tuning, is often taken to optimize DNNs. In the pre-training stage, restricted Boltzmann machines (RBMs) are generatively learned and stacked layer-by-layer, producing a multilayer generative model called deep belief network (DBN). In the fine-tuning stage, a final soft-max layer is added to the DBN and the whole network is tuned discriminatively through error backpropagation. This kind of network is referred to as DBN-DNN for clarity.

One shortcoming of the DBN-DNN is that, the DBN is pre-trained using a greedy bottom-up pass [4]. The top-down influences on the inference process are ignored. Parameters of the lower-level layers are never adjusted when pre-training one layer of the network, thus the pre-trained DBN is suboptimal.

Recently, new learning algorithms for deep Boltzmann machines [5] were proposed. DBMs retain the many-layer deep architecture of DBNs, the unsupervised generative pre-training procedure, and have the advantages that a top-down feedback pass is incorporated, and the parameters of all layers can be jointly optimized [5, 6]. DBMs can also be used for classification tasks by adding a top soft-max layer. It is referred to as DBM-DNN in our work.

DBM-DNNs have been successfully applied to tasks such as handwritten digit recognition, object recognition [5, 7], and spoken query detection [8]. In this paper, we attempt to use DBM-DNN for acoustic modeling. The DBM-DNN acoustic models are examined on the TIMIT phone recognition task. A variety of configurations of the DBM-DNN models are investigated and discussed. Moreover, the dropout fine-tuning method, which was proposed most recently [9], is also investigated in our DBM-DNN acoustic models.

The rest of the article is organized as follows. Section 2 briefly describes how to construct and train a DBM. The experimental results are reported and analyzed in Section 3. Our conclusions are summarized in Section 4. In Section 5, we provide a discussion of relation to prior work.

2. TRAINING DBM-DNN MODELS

Before introducing the learning algorithm of DBM-DNNs, we provide an overview of the model architecture. We use a two-hidden layer DBM-DNN to present the differences of the architecture comparing with DBN-DNN. The architecture of the two-hidden-layer DBM-DNN is depicted in figure 1. Training the DBM involves the top-down feedback when inferring the hidden layer $\mathbf{h}_1$, which leads to the novel architecture of an additional input in the DBM-DNN model. The rest of this section focuses on how to train the DBM-DNN model.

This work was supported by 863 program in China (No. 2011AA01A207), Beijing Natural Science Foundation (No.4132071) and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

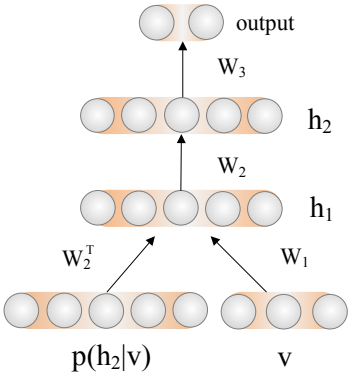

Fig. 1. The DBM-DNN model. The posterior $p(\mathbf{h}_2|\mathbf{v})$ are used as additional input.

Like DBNs, DBMs are also probabilistic generative models. The learning algorithm proposed by Salakhutdinov [5] provides a new way to train DBMs. The energy function of the two–hidden-layer DBM model is defined as(ignoring bias terms):

$$E(\mathbf{v}, \mathbf{h}_1, \mathbf{h}_2; \theta) = -\mathbf{v}^T \mathbf{W}_1 \mathbf{h}_1 - {\mathbf{h}_1}^T \mathbf{W}_2 \mathbf{h}_2 \quad (1)$$

where $\mathbf{W}_1$, $\mathbf{W}_2$ are the weight matrices between visible-to-hidden and hidden-to-hidden layers. The probability that the DBM model assigns to visible vector $\mathbf{v}$ can be obtained through energy function:

$$p(\mathbf{v}; \theta) = \frac{\sum_{\mathbf{h}_1, \mathbf{h}_2} \exp(-E(\mathbf{v}, \mathbf{h}_1, \mathbf{h}_2; \theta))}{\sum_{\mathbf{v}} \sum_{\mathbf{h}_1, \mathbf{h}_2} \exp(-E(\mathbf{v}, \mathbf{h}_1, \mathbf{h}_2; \theta))} \quad (2)$$

From formula 2, the gradient of negative log-likelihood is obtained in the form of

$$-\frac{\partial \log \mathcal{L}}{\partial \theta} = \langle \frac{\partial E}{\partial \theta} \rangle_{data} - \langle \frac{\partial E}{\partial \theta} \rangle_{model} \quad (3)$$

where $\langle \frac{\partial E}{\partial \theta} \rangle_{data}$ and $\langle \frac{\partial E}{\partial \theta} \rangle_{model}$ denote data-dependent expectation and model's expectation respectively.

Exact calculation of the second part of formula 3 takes exponential time and is intractable. To address this problem, Salakhutdinov [5] proposed an efficient approximate learning algorithm. The data-dependent expectations are estimated by using the mean-field inference and the model's expected statistics are estimated by using Markov-chain Monte-Carlo (MCMC) sampling (for more details see [5, 7]).

The initial model parameters θ can be obtained by using modified RBMs [5]. In the stage of training these RBMs, the influences of the top-down and bottom-up pass require special consideration. For the first RBM, when inferring the hidden layer $\mathbf{h}_1$, the input are doubled. Conversely, for the last RBM, the activations of hidden layer are doubled. The whole pre-training process does not require any supervised information.

The above procedures describe binary-binary DBMs. For speech recognition tasks, the input $\mathbf{v}$ is continuous. We use Gaussian-Bernoulli DBMs [10].

After obtaining the initial parameters, the mean-field and MCMC approximation procedure is used to jointly optimize the whole DBM. Then, the optimized DBM model is used to initialize a DNN. Furthermore, the DBM-DNN model is discriminatively fine-tuned by using standard back-propagation. The model parameter $\mathbf{W}_3$ is initialized randomly.

3. EXPERIMENTS

In this section, a variety of configurations of the DBM-DNN models are investigated and discussed, including different layer numbers, layer sizes, limited amount of labeled data, and the dropout fine-tuning method. Results are also compared with DBN-DNN models.

3.1. Experimental setup

All experiments are conducted on the TIMIT corpus to evaluate the DBM-DNN acoustic models for phone recognition. The 3696 sentences from 462 speakers are used for training. A development set of 50 speakers is selected from the test set. That set does not overlap with the core set. The 24-speaker core test set is used for evaluation.

The conventional 13-dimension MFCC features, along with their first and second derivatives are used. Cepstral mean and variance normalization is performed on per utterance case. For DNN models 11 consecutive frames are used as network input. The baseline acoustic model is a 61-phone context independent GMM-HMM model, which is trained using HTK in maximum likelihood fashion. To train DNN models, the corpus is labeled using forced alignment with this baseline model.

All decoding experiments use the same bigram phone language model and same decoding parameters, only acoustic models are changed. For scoring the recognition results are mapped to 39 units [11] after decoding.

3.2. Different configurations of DBM-DNNs

In the first experiment, the DBM-DNNs with various hidden layer numbers and layer sizes are compared. The same training recipe is used for all the configurations. First pre-trained RBMs are stacked as the initial parameters of a DBM. Then in the inference procedure of the DBM, for each mini-batch of training data, 3 Gibbs updates in the MCMC procedure and 5 iterations of mean-field inference are used. The learning rate is set to 0.01 initially and is multiplied by 0.99998 after each mini-batch. The inference procedure is stopped after 100 iterations. After that a randomly initialized soft-max layer with 183 outputs are added on top of the pre-trained DBM. Standard back propagation is used to fine-tune the DBM-DNN model. For both pre-training and fine-tuning the mini-batch size is all set to 128.

The phone error rate results are listed in table 1. The baseline GMM gives PER of 29.54% on the core set, and

Table 1. *PER of various acoustic models. axb in the column of configurations means a deep model with a hidden layers and each has b units.*

acoustic model	configurations	devset	coreset
GMM	-	28.69%	29.54%
DBM-DNN	2x512	22.98%	23.93%
	2x1024	22.02%	23.10%
	2x1536	21.38%	22.99%
	2x2048	**21.46%**	**22.62%**
	3x1024	22.37%	23.42%
DBN-DNN	3x1024	21.60%	23.40%
	3x2048	21.86%	23.51%
	7x1024	21.51%	22.76%
	7x2048	21.30%	23.02%

the 2x2048 DBM-DNN model achieves 22.62%. The relative PER reduction is 23.4%, which indicates the more powerful modeling ability of DBM-DNNs. Comparing the PER results of different configurations, we can see that it benefits from increasing the layer size. By increasing layer size from 512 to 2048, the PER is reduced by 1.5% and 1.3% absolute on the development set and the core set. But it results in worse PER when increasing layer number from 2 to 3. The reason may lie in the DBM training procedure, because it does not guarantee to improve the variational bound by adding an extra layer. Similar results were reported in [5, 8].

The results of DBN-DNN acoustic models are also given in table 1. All the DBN-DNN models are trained following the learning schedule in [1]. When same amount of parameters are used, the 2x2048 DBM-DNN outperforms the 3x2048 DBN-DNN. (The DBM-DNN has extra 2048 input units. For more details see section 2.) The PER is reduced by 3.8% relatively on the core test set. Even comparing with a deeper DBN-DNN model with 7 hidden layers, the DBM-DNN is still slightly better. We believe that incorporating a top-down feedback in the approximate inference procedure is the critical factor that contributes to the better recognition accuracies from DBM-DNN models. It makes DBM-DNN more robust to deal with ambiguous inputs, which generally exist in the features of speech signals.

3.3. Partially labeled data training

Since DBM-DNNs with fewer amounts of parameters achieve same accuracy as DBN-DNNs, they may be more useful for low resource tasks. This inspires us a further investigation when limited amount of labeled data are available. In the second experiment, during the fine-tuning stage of the 2x2048 DBM-DNN and the 3x2048 DBN-DNN, different ratios of labeled data are randomly selected from training corpus. The initial weights are from the same pre-trained model as in section 3.2.

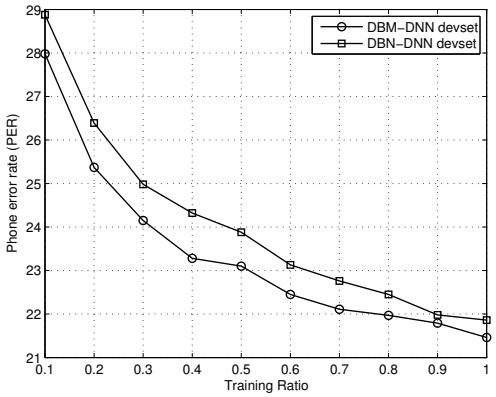

Fig. 2. Comparison of DBM-DNN with DBN-DNN in terms of PER on TIMIT development set using different training ratios.

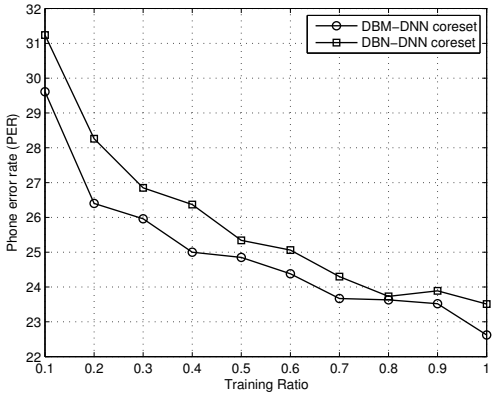

Fig. 3. Comparison of DBM-DNN with DBN-DNN in terms of PER on TIMIT core set using different training ratios.

It is shown in figure 2 and 3 that the DBM-DNN consistently outperforms the DBN-DNN. The performance margin is larger when the amount of labeled data is fewer than 40%. When 20% labeled data are used, the DBM-DNN gets 6.6% relative PER reduction comparing with the DBN-DNN model. We can also see that using only 10% of labeled data the DBM-DNN model achieves comparable performance to the baseline GMM model (29.61% of DBM-DNN **v.s.** 29.54% of GMM on the core test set). This interesting result inspires us to apply DBM-DNN models for low resource tasks in the future. Moreover, using partially labeled data significantly reduces calculation loads since the back-propagation procedure is very computational intensive. This may be helpful when a system is required to be built quickly.

3.4. Dropout fine-tuning

Generally, neural networks with many hidden units and deep architecture can get great performance on training set but do worse on test data if there is only a limited amount of training data. The conventional way to improve the performance

on test data is averaging the output from a large scale number of different networks. However, training these networks is impractical in a reasonable time and accomplishing the test with a large number of networks is computationally expensive. The dropout procedure [9] provides an efficient way to perform the work of averaging different networks. Moreover, the dropout procedure is performed with a single network thus the testing is more efficient than conventional model averaging approaches.

In this experiment, dropout fine-tuning is performed with the standard, stochastic gradient descent procedure. We use the 2x2048 DBM-DNN and 50% dropout for hidden layers. Mini-batch size of 128 is used for dropout-backpropagation. A small constant learning rate of 0.008 is used. We apply the total gradient on a mini-batch. The model is trained for 200 epochs to converge. The best results show that PER is 21.46 on the core test set and 19.51 on the development set. The DBM-DNN model gets additional relative PER reduction of 5.1% on the core test set when the standard back-propagation is replaced by dropout-backpropagation.

4. CONCLUSIONS

In this paper, we have successfully applied DBM-DNNs on acoustic modeling. The experimental results on TIMIT showed that the 2x2048 DBM-DNN can achieve 23.4% relative error reduction on the core test set when comparing with the baseline GMM model. Using same amount of parameters, DBM-DNN models perform better than DBN-DNN models. Moreover, DBM-DNN models more obviously outperform DBN-DNN models when using only very limited amount of labeled data. Applying the dropout strategy obtains another 5.1% relative error reduction on the core test set.

The two main challenges we are facing are to improve the efficiency of training to perform the DBM-DNN acoustic modeling on larger data sets, and to develop effective algorithms for training DBMs with more hidden layers.

5. RELATION TO PRIOR WORK

To our knowledge, the work of this paper is the first time that DBM-DNNs are applied on speech recognition. Our work is based on the efficient learning algorithm of DBMs proposed by [5].

The work of [5] focus on the learning algorithms of DBMs and the application on handwritten digit recognition and object recognition. The work of [8] uses DBMs for spoken query detection, where DBMs are used to generate posteriorgrams. In our work DBMs are used as acoustic models in a phone recognition system. In our work we investigate different configurations of DBM-DNN acoustic models. The recently proposed dropout fine-tuning is also incorporated in our experiments.

The work of [1, 2] use DBN-DNN for acoustic modeling, while our work investigates DBM-DNN, which is a new type of DNNs, and has the advantage of jointly optimization of the whole network.

6. REFERENCES

[1] Abdel rahman Mohamed, George E. Dahl, and Geoffrey Hinton, "Acoustic Modeling Using Deep Belief Networks," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, pp. 14–22, 2012.

[2] Frank Seide, Gang Li, and Dong Yu, "Conversational speech transcription using context-dependent deep neural networks," in *INTERSPEECH*, 2011, pp. 437–440.

[3] Tara N. Sainath, Brian Kingsbury, Bhuvana Ramabhadran, Petr Fousek, Petr Novák, and Abdel rahman Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *ASRU*, 2011, pp. 30–35.

[4] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[5] Ruslan Salakhutdinov and Geoffrey E. Hinton, "Deep boltzmann machines," *JMLR*, vol. 5, pp. 448–455, 2009.

[6] Ruslan Salakhutdinov and Hugo Larochelle, "Efficient learning of deep boltzmann machines," *JMLR*, vol. 9, pp. 693–700, 2010.

[7] Ruslan Salakhutdinov and Geoffrey Hinton, "An Efficient Learning Procedure for Deep Boltzmann Machines," *Neural Computation*, vol. 24, pp. 1967–2006, 2012.

[8] Yaodong Zhang, Ruslan Salakhutdinov, Hung-An Chang, and James R. Glass, "Resource configurable spoken query detection using deep boltzmann machines.," in *ICASSP*, 2012, pp. 5161–5164.

[9] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[10] KyungHyun Cho, Alexander Ilin, Tapani Raiko, and Tapani Raiko, "Improved learning of gaussian-bernoulli restricted boltzmann machines.," in *ICANN*, 2011, pp. 10–17.

[11] Kai fu Lee and Hsiao wuen Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 1641–1648, 1989.