Modular Combination of Deep Neural Networks for Acoustic Modeling

Jonas Gehring[1], *Wonkyum Lee*[3], *Kevin Kilgour*[1], *Ian Lane*[3], *Yaije Miao*[2], *Alex Waibel*[1,2]

[1] Interactive Systems Lab, Karlsruhe Institute of Technology; Germany
[2] Language Technologies Institute, Carnegie Mellon University; Pittsburgh, PA; USA
[3] Silicon Valley Campus, Carnegie Mellon University; USA

jonas.gehring@kit.edu wonkyum.lee@sv.cmu.edu

Abstract

In this work, we propose a modular combination of two popular applications of neural networks to large-vocabulary continuous speech recognition. First, a deep neural network is trained to extract bottleneck features from frames of mel scale filterbank coefficients. In a similar way as is usually done for GMM/HMM systems, this network is then applied as a non-linear discriminative feature-space transformation for a hybrid setup where acoustic modeling is performed by a deep belief network. This effectively results in a very large network, where the layers of the bottleneck network are fixed and applied to successive windows of feature frames in a time-delay fashion. We show that bottleneck features improve the recognition performance of DBN/HMM hybrids, and that the modular combination enables the acoustic model to benefit from a larger temporal context. Our architecture is evaluated on a recently released and challenging Tagalog corpus containing conversational telephone speech.

Index Terms: Acoustic Modeling, Deep Belief Networks, Deep Bottleneck Features, Large Vocabulary Speech Recognition

1. Introduction

Recently, multiple works have demonstrated that the performance of automatic speech recognition systems can be heavily improved by using deep neural networks (DNNs) for acoustic modeling [1], [2]. The key advantages over much earlier approaches [3] to this hybrid setup combining neural networks and hidden Markov models are improved learning algorithms that can leverage the high modeling capacity of deep networks [4] and the usage of a large number of context-dependent phonetic target states during network training.

Work is still done in determining which speech features are most useful when training neural network acoustic models. For generative models like restricted Boltzmann machines, it has been argued that raw mel scale spectral coefficients are more suitable than further preprocessed features with reduced covariances like MFCCs [5]. In practice, though, it appears that deep networks can been trained with similar performance on a variety of acoustic data, including windows of features reduced with linear discriminant analysis or speaker-adapted features [6].

For the standard approach to ASR, in which Gaussian mixtures are used for acoustic modeling, a large amount of research regarding input feature engineering has already been done. In particular, neural networks have been employed for feature extraction in the form of tandem features [7] or bottleneck features [8]. There, the activations of either the output layer or a narrow hidden layer, respectively, are used as input features after the corresponding neural network has been trained to estimate phonetic target states from windows of acoustic data.

In this work, we want to investigate whether bottleneck features are useful for acoustic modeling with deep networks as well. As in recognition systems using Gaussian mixtures, we extract bottleneck features from several adjacent windows of speech features. This way, the estimation of phoneme states is effectively performed by a combination of two deep neural networks, where the activations of the first network's small hidden layer from different time-stamps are the input for the second one.

2. Related Work

The motivation for the proposed architecture is two-fold. For one, the modular combination of separately trained sub-networks into bigger networks is a well-established design principle when building large classifiers. Second, bottleneck features can be regarded as a probabilistic and discriminative dimensionality reduction technique that is known to work well with GMM/HMM system, and they might be used to improve neural network acoustic models, too.

The idea of using sub-components of simple neural networks to build networks with higher complexity for difficult speech recognition has been extensively explored in the past. Waibel proposed methods for modular construction of networks for phoneme recognition almost 25 years ago, and incorporated the hidden units of networks trained to discriminate between few classes into a new network that was then trained to detect a superset of the original classes [9]. This architecture included time-delay units to account for variability in the temporal domain of speech signals, in a similar way to what is commonly known as convolutional neural networks today.

More recent applications of similar techniques to large-vocabulary speech recognition include hierarchical combinations of tandem features [10] as well as bottleneck features [11] to design more powerful preprocessors of acoustic data for GMM/HMM systems. However, as of yet, no work has been done on adopting those ideas for hybrid DNN/HMM systems used in LVCSR tasks.

From a feature engineering point of view, other, non-probabilistic dimensionality reduction techniques have been evaluated for DNN acoustic modeling. Mohamed et al. performed linear discriminant analysis (LDA) on frames of MFCC features, followed by vocal-tract length normalization (VTLN) and speaker-adaptation using a maximum likelihood linear regression performed in feature space (fMLLR) [12]. They reported no improvement by training a network on LDA-transformed features on the TIMIT benchmark for phone recognition. With speaker-adapted features, significant gains of a

similar magnitude as for GMM-based systems could be obtained. Corresponding results were obtained by Seide et al. on a large-vocabulary conversational speech recognition task [6]. In particular, they found that neither heteroscedastic LDA nor VTLN improved the recognition accuracy of their CD-DNN-HMM architecture, but they achieved gains by applying an fMLLR-like transform for adaptation.

In contrast to those works, we propose the usage of deep bottleneck features obtained from log mel scale filterbank coefficients instead of performing an LDA transform on standard input features. This combination of two deep neural networks improves recognition performance and enables the network to leverage an increased temporal context of speech features.

3. Deep Bottleneck Features

For extracting bottleneck features from standard speech features, we use the deep bottleneck architecture as described in our earlier work [13]. This is an extension of standard bottleneck features commonly used in automatic speech recognition, in which a neural network containing a narrow hidden layer is trained to predict phone states [8]. The activations obtained in the small hidden bottleneck layer are then used as input features for a standard GMM-based recognition system. Since the number of hidden units in the narrow layer are usually much smaller than the dimensionality of the network input, this approach can be viewed as a probabilistic and discriminative dimensionality reduction.

In [13], a deep neural network is constructed by placing a stack of pre-trained denoising auto-encoders in front of the bottleneck. Denoising auto-encoders try to reconstruct the original version of input data corrupted with random noise, which enables them to learn over-dimensional representations without degrading to trivial solutions for their parameters [14]. This is suitable for bottleneck extraction networks, where the layers preceding the bottleneck are usually much wider than the network input.

The general training procedure can be summarized as follows. First, a stack of denoising auto-encoders is unsupervised pre-trained on frames of stacked log mel scale coefficients, following the layer-wise training procedure initially proposed by Vincent et al [14]. The encoding part of each layer is then used to initialize a deep neural network, and randomly initialized layers are added on top for the bottleneck, an additional non-linear transformation and the network output. Finally, supervised fine-tuning is performed on the whole network using the phone states assigned to the input frames.

4. Deep Belief Network Acoustic Modeling

4.1. Restricted Boltzmann Machines

In this work, we use deep belief networks (DBNs), a particular type of deep neural networks, for acoustic modeling. A DBN consists of multiple stacked restricted Boltzmann machines (RBMs), each being pre-trained in an unsupervised manner on the actual input features or the hidden representation of the previous one [4]. RBMs are bipartite graphical models in which hidden units learn a representation of visible units. In the standard configuration, both visible and hidden units are binary units that are sampled from a Bernoulli distribution. The probability of being active is computed using weighted connections to the hidden and visible units, respectively.

RBMs are energy-based models, and each configuration of visible units $\boldsymbol{v}$ and $\boldsymbol{h}$ is assigned an energy term E:

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i=1}^{V}\sum_{j=1}^{H} v_i h_j w_{ij} - \sum_{i=1}^{V} v_i c_i - \sum_{j=1}^{H} h_j b_j$$

where w_{ij} is the weight assigned to the connection between a visible unit v_i and h_j, and c_i and b_j are their bias terms. For modeling real-valued data, which is the usual case for acoustic features, the binary visible units can be replaced with Gaussian units [15]. The energy of a configuration becomes:

$$E(\boldsymbol{v}, \boldsymbol{h}) = \sum_{i=1}^{V} \frac{(v_i - c_i)^2}{2\sigma^2} - \sum_{j=1}^{H} b_j h_j - \sum_{i=1}^{V}\sum_{j=1}^{H} \frac{v_i}{\sigma} h_j w_{ij}$$

with σ representing the variance of the normal distribution from which the visible units are sampled.

Unsupervised learning of a model is done by maximizing the log-likelihood for known configurations (i.e., the training data) as determined by the energy term. The contrastive divergence algorithm provides a fast approximation of this objective by computing the difference of correlations between two configurations obtained by alternating Gibbs sampling [16]. For further details about training RBMs, the interested reader is referred to [17].

After pre-training a stack of RBMs, the weights and biases of the hidden units can be used to initialize the hidden layers of a deep belief neural network. When used for discriminative training, an additional classification layer is connected to the last hidden layer, and the resulting network is fine-tuned with standard backpropagation.

4.2. Acoustic Modeling

When employing neural networks as acoustic models in combination with hidden Markov models, they are used to compute a posteriori emission probabilities of phone states [3]. If the network is trained to estimate probabilities $p(q_t|\boldsymbol{x}_t)$ of states q_t given observations as input feature vectors $\boldsymbol{x}_t$ using a cross-entropy criterion, the emission probabilities can be obtained with Bayes' rule:

$$p(\boldsymbol{x}_t|q_t) = \frac{p(q_t|\boldsymbol{x}_t)p(\boldsymbol{x_t})}{p(q_t)}$$

where $p(q_t)$ denotes the prior probability of a phone state, which is estimated using the available training data. During decoding, the most likely sequence of states is computed by the HMM. Since the observation $\boldsymbol{x}$ is independent of the state sequence, its probability $p(\boldsymbol{x_t})$ can be ignored.

4.3. Modeling Bottleneck Features

In this work, we use deep belief networks to model a window of bottleneck features, extracted by applying the respective neural network to adjacent windows of acoustic features as illustrated in Fig. 1. This extraction scheme is related to the frame shifting done with individual layers in time-delay neural networks [9] and forms the basis of many hierarchical or convolutional architectures. The weights of the bottleneck network are fixed during DBN training, so that each network is trained in isolation. In practice, this means that acoustic model training can be accelerated if training examples are generated only once by computing bottleneck features for all available training data.

In our setup, bottleneck features could be regarded as binary features, since the units in the bottleneck layer use a sigmoid activation function. Here, we choose to model them as

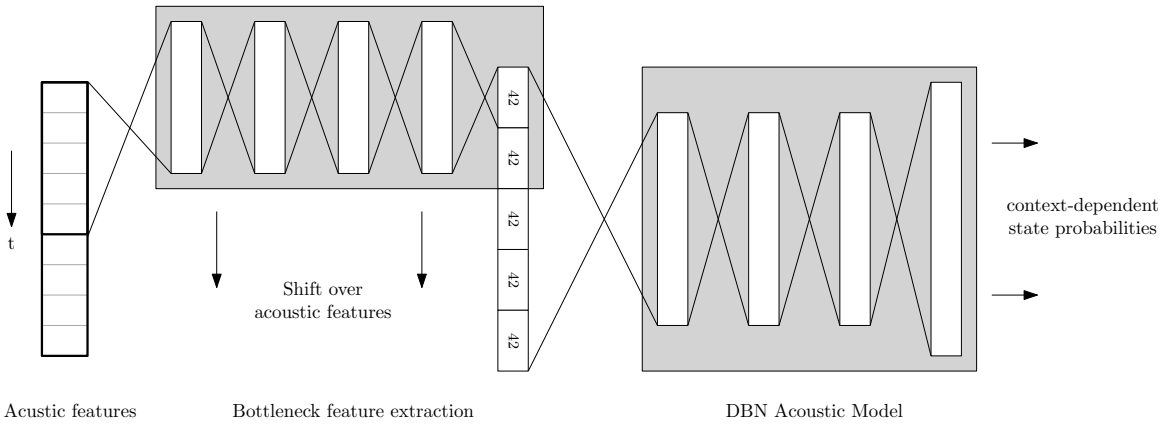

Figure 1: Proposed architecture for acoustic modeling, illustrating a bottleneck extraction network consisting of 4 hidden layers and a deep belief network estimating class probabilities with 3 hidden layers.The first network is computed at 5 positions over windows of 5 acoustic feature vectors, resulting in a window size of 9 feature vectors for the acoustic model.

real-valued data by using their actual activation value without applying the sigmoid non-linearity. This serves to purposes: first, the same DBN architecture can be used on log mel scale features as well as BNFs, which helps in comparing the resulting recognition performances. Second, this allows for easy integration of further enhancements like fMLLR training or linear bottlenecks [18] that all provide real-valued features.

5. Experimental Setup

5.1. Corpus and Baseline System Description

We trained our systems on a Tagalog dataset which was recently released as the IARPA Babel Program Tagalog language collection babel106-v0.2f [19]. It contains 79 hours of conversational telephone speech, from which 69 are used for training and 10 for testing the recognition systems.

For the GMM/HMM baseline system, we used MFCCs computed from 30 log mel scale filterbank coefficients, which were in turn extracted from audio data by applying a hamming window with a length of 20 ms and a frame shift of 10 ms. 13 MFCCs were concatenated with 10 contextual samples, forming feature vectors containing 143 elements. After performing per-speaker cepstral mean and variance normalization, the final feature vector consisted of 42 elements obtained by applying LDA. During ML-training, 4000 context-dependent states with an average of 53 Gaussian components per state were used.

5.2. Deep Bottleneck Features Training

The neural network for extracting deep bottleneck features (DBNFs) was trained as described in [13]. 30 filterbank coefficients were obtained as described above, normalized on a per-speaker level and concatenated with past and future samples to feature frames consisting of 330 elements. Five auto-encoder layers containing 1000 units each were stacked and pre-trained individually, and a bottleneck layer with 42 units as well as one additional hidden layer and a classification layer were added. The network was then trained to predict context-independent monophone states. A random subset containing 5% of the available training examples was used as a held-out validation set to perform early stopping.

The GMM/HMM systems trained on bottleneck features used the phonetic decision tree from the MFCC baseline and therefore ended up with the same number of tied states. As for the baseline, features extracted from 11 adjacent positions were reduced to 42 dimensions with LDA. Speaker-adaptive training was performed using fMLLR. The DBNF system was used to generate new alignments for training the deep belief network acoustic models.

5.3. Acoustic Model Training

For the DBN acoustic models, RBMs were pre-trained layer-wise and unsupervised with the contrastive divergence algorithm, following the recommendations in [17]. Input data was given by 40 log mel filterbank coefficients (lMEL) or 42 bottleneck features concatenated to varying window sizes. In the first layer, a RBM with Gaussian visible units was trained for 10 epochs using stochastic gradient descent with mini-batches containing 128 examples. The learning rate was linearly decayed over the total training time from $5 \cdot 10^{-4}$ to $1 \cdot 10^{-4}$, and the gradients obtained were smooth using a momentum term of 0.5. For the other layers, standard RBMs with binary visible units were trained for 5 epochs with learning rates scaled by a factor of 10.

All RBMs were trained with a sparsity constraint proposed by Nair and Hinton [20], which was found to improve frame-level classification as well as final recognition performance. The gradients obtained using contrast divergence were augmented with gradients from a cross-entropy sparsity cost, which compares an exponentially decaying average of mean activations of the hidden units to a small target value.

Finally, supervised training was performed for 25 epochs with a batch size of 128, with a linearly decaying learning rate with decreased from 0.1 to 0.001 for the first 20 epochs. As for pre-training, a momentum term of 0.5 was used.

The network architecture, the hyper-parameters described above as well as the final decoding parameters were optimized on log mel scale data. We settled with 5 stacked RBMs containing 2000 units each. The number of context-dependent target states for supervised training set to the number of tied states in the respective baseline systems. Pre-training and fine-tuning of all neural network models was implemented with the Theano library [21].

Acoustic Model	Features	Window	WER
GMM	MFCC	11	69.7
	DBNF	21	59.6
	DBNF+fMLLR	21	56.6
DBN	lMEL	11	58.1
		21	54.8
		31	54.7
		41	55.1
	DBNF	21	53.0
		31	52.0
		41	52.6

Table 1: Recognition performance for the different systems described. The window column contains the effective number of feature vectors accessible to the acoustic model.

6. Results

Table 1 lists the recognition performances in terms of word error rate (WER) for the baseline system with and without deep bottleneck features as well as for DBN/HMM systems with varying features and effective window sizes. Regarding the performance of the baseline system, it should be noted that those systems are still among the early system builds for this fairly recent corpus. It can be seen that DBNFs yield high gains in accuracy, lowering the baseline WER to 59.6% (-14.5% relative). Applying fMLLR training produces further improvements (-5% relative).

The hybrid DBN/HMM combination outperforms the speaker-adaptive bottleneck feature setup, resulting in relative improvements of up to 21.5% (54.7% WER) over the MFCC baseline. It can be seen that a window of at least 21 features is required to obtained good recognition accuracy. Further enlargement of the window produces only minuscule gains, and with 41 feature vectors performance is being degraded.

When using bottleneck features for training the neural network acoustic models, improvements over log mel scale input are obtained. The best result of 52.0% WER on a window of 31 feature vectors marks a relative improvement of nearly 5% over the best lMEL setup and a 25% improvement over the baseline system.

The results also show that the network trained on bottleneck features benefits from an increased temporal context in that increasing the window size from 21 to 31 has a notable effect on recognition performance (-2.3% relative), which stands in contrast to the DBNs trained on log mel data where only diminishing improvements are obtained. However, increasing the number of input feature vectors to 41 results in slightly worse performance for both systems.

7. Conclusions

With the results obtained above, we have demonstrated that bottleneck features are useful input features for DBN/HMM speech recognition setups. It could be shown that the modular combination proposed enables the acoustic model to use an increased temporal context of acoustic features more efficiently than an identical network trained directly on the input features.

The performance improvements achieved by using deep bottleneck features for the hybrid DBN/HMM systems are significant, though not as large as for the GMM/HMM baseline system. However, deep neural networks have a much higher modeling capacity then GMMs, and it is to be expected that a good part of the modeling performed in the bottleneck network can be learned in a standalone DBN as well. Nevertheless, we regard our approach to combining neural networks for acoustic modeling as promising and the general principles of modularity as an important paradigm that is applicable to deep neural networks as it is to shallow ones.

For the specific architecture proposed, more experiments may be desirable in order to obtain greater insight into the interplay between feature extraction and acoustic modeling. Furthermore, future work will deal with integrating DBNFs and DBN training into a single model, so that joint fine-tuning of the whole network is possible after its individual components have been optimized. It will be interesting to investigate in how to perform efficient speaker-adaptive training on the bottleneck feature level as well.

8. Acknowledgements

Supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

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, 2012.

[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, 2011, pp. 437–440.

[3] N. Morgan and H. Bourlard, "Continuous speech recognition using multilayer perceptrons with hidden markov models," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. IEEE, 1990, pp. 413–416.

[4] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[5] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4273–4276.

[6] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 24–29.

[7] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 3. IEEE, 2000, pp. 1635–1638.

[8] F. Grézl and P. Fousek, "Optimizing bottle-neck features for LVCSR," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4729–4732.

[9] A. Waibel, "Modular construction of time-delay neural networks for speech recognition," *Neural computation*, vol. 1, no. 1, pp. 39–46, 1989.

[10] F. Valente, M. M. Doss, C. Plahl, and S. Ravuri, "Hierarchical processing of the modulation spectrum for gale mandarin lvcsr system," *Proc. Interspeech09*, 2009.

[11] F. Grézl and M. Karafiát, "Hierarchical neural net architectures for feature extraction in asr," *Proc. INTERSPEECH10*, pp. 1201–1204, 2010.

[12] A.-R. Mohamed, T. N. Sainath, G. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, "Deep belief networks using discriminative features for phone recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5060–5063.

[13] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, to appear.

[14] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[15] A.-r. Mohamed, G. E. 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, 2012.

[16] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[17] G. Hinton, "A practical guide to training restricted boltzmann machines," University of Toronto, Machine Learning Group, Tech. Rep. UTML TR 2010-003, 2010.

[18] K. Vesely, M. Karafiát, and F. Grézl, "Convolutive bottleneck network features for lvcsr," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 42–47.

[19] "IARPA, Office for Incisive Analysis, Babel Program," http://www.iarpa.gov/Programs/ia/Babel/babel.html, IARPA, retrieved 2013-03-06.

[20] V. Nair and G. Hinton, "3d object recognition with deep belief nets," *Advances in Neural Information Processing Systems*, vol. 22, pp. 1339–1347, 2009.

[21] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU Math Expression Compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Jun. 2010, Oral Presentation.