AUTO-ENCODER BOTTLENECK FEATURES USING DEEP BELIEF NETWORKS

Tara N. Sainath, Brian Kingsbury, Bhuvana Ramabhadran

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598
{tsainath, bedk, bhuvana}@us.ibm.com

ABSTRACT

Neural network (NN) bottleneck (BN) features are typically created by training a NN with a middle bottleneck layer. Recently, an alternative structure was proposed which trains a NN with a constant number of hidden units to predict output targets, and then reduces the dimensionality of these output probabilities through an auto-encoder, to create auto-encoder bottleneck (AE-BN) features. The benefit of placing the BN after the posterior estimation network is that it avoids the loss in frame classification accuracy incurred by networks that place the BN before the softmax. In this work, we investigate the use of pre-training when creating AE-BN features. Our experiments indicate that with the AE-BN architecture, pre-trained and deeper NNs produce better AE-BN features. On a 50-hour English Broadcast News task, the AE-BN features provide over a 1% absolute improvement compared to a state-of-the-art GMM/HMM with a WER of 18.8% and pre-trained NN hybrid system with a WER of 18.4%. In addition, on a larger 430-hour Broadcast News task, AE-BN features provide a 0.5% absolute improvement over a strong GMM/HMM baseline with a WER of 16.0%. Finally, system combination with the GMM/HMM baseline and AE-BN systems provides an additional 0.5% absolute on 430 hours over the AE-BN system alone, yielding a final WER of 15.0%.

***Index Terms*—** Deep Belief Networks, Speech Recognition

1. INTRODUCTION

Artificial neural networks (ANNs) [1] are a popular acoustic modeling technique in speech recognition systems. Perhaps the most popular ANN to date is the multi-layer perceptron (MLP), which organizes non-linear hidden units into layers and has full weight connectivity between adjacent layers. During training, these weights are initialized with small random values. MLPs are used to estimate a set of state-based posterior probabilities, which are used in a variety of ways. In a hybrid system [1], these probabilities are used as output probabilities of a Hidden Markov Model (HMM). Alternatively, approaches such as TANDEM [2] and bottleneck features (BN) [3] derive a set of features from the MLP, which are then used as input features into a traditional Gaussian Mixture Model (GMM)/HMM system. Typically, TANDEM and BN methods have performed better relative to hybrid systems.

One reason for this is that MLP training is initialized from random weights and the objective function is non-convex, so training can get stuck in a poor local optimum. While this has not been a serious hindrance for MLPs having one or two hidden layers that are trained using stochastic gradient descent [4], it poses a more serious challenge for deeper MLPs. Recently, Restricted Boltzmann Machines (RBMs) [5] have been explored to pre-train weights of ANNs in an unsupervised fashion. This pre-training allows for a much better initial weight estimate, addressing the problems with MLP training. An ANN which is pre-trained with RBMs is generally referred to as a deep belief network (DBN). When DBNs are used in a hybrid architecture, they have shown promising results for various ASR tasks [6], [7]. Recently, extracting BN features from DBNs, has also shown improvements over extracting bottleneck features from randomly initialized MLPs [8].

The most common approach to extract BN features is to train a DBN with a narrow bottleneck middle layer. For example, a typical network topology could be 360-1024-40-1024-384, where 360 is the dimensionality of the input feature, 1024 is the number of hidden units, and 384 is the number of output targets. This architecture is used in many ways in the literature [3], [8]. In [3], raw input features (i.e., TRAPS) with a large temporal context are input into the MLP when deriving BN features. The intuition is that these features are often complementary to typical short-time speech features (i.e., MFCCs, PLPs), and therefore concatenating the BN and original speech features offer improvements over using the original speech features alone. Furthermore, in [8], speech features are used as input into a pre-trained DBN in order to extract BN features, though the DBN hybrid system outperforms the BN system.

The goal of BN features is to derive a set of features which capture information about the good classification accuracy at the output targets. We argue that placing a BN layer in the middle of the DBN degrades the frame accuracy of the output targets, and therefore the full benefit of the BN features cannot be achieved. This results in DBN systems used to extract BN features performing better than the BN systems themselves [8], or using BN features in tandem [3] rather than by themselves.[1]

Alternatively, [10] trained an MLP without a BN layer, for example 360-1024-1024-384. After the MLP is finished training, an auto-encoder neural network is trained with a BN layer, to reduce the dimensionality of the 384 output targets to 40. The auto-encoder bottleneck (AE-BN) structure proposed in [10] allowed for gains when combined with other systems on a large 1,800 hour GALE Arabic task. The benefit of the AE-BN approach is twofold. First, the AE-BN method ensures that the best achievable frame accuracy at the output targets is obtained before further reducing dimensionality. Second, the AE-BN architecture allows output targets built from NNs coming from different feature streams to be linearly combined before extracting a bottleneck layer, something which the typical BN architecture [3] does not allow for. In this paper, we show that when using typical speech features as input to the DBN, having pre-trained and deeper networks help to improve the AE-BN further. This allows the AE-BN features alone to provide benefits over hybrid DBN and GMM/HMM systems trained from the same input speech features.

Our initial experiments are conducted on a 50-hour English

[1] The authors are aware that [9] proposed a BN architecture which provided gains over regular speech features when not used in tandem. However, this approach looked at modeling the context of raw features, something we do not consider in this work.

Broadcast News task [11]. First, we show that pre-trained and deeper networks which allow for improvements in hybrid DBN systems also improve the AE-BN features. Second, we show that using AE-BN features alone offer a 1.3% absolute improvement over a state-of-the-art [7] speaker-adapted, discriminatively trained GMM/HMM baseline and 0.9% absolute improvement over a hybrid DBN system. To our knowledge, this is the first use of bottleneck features to offer improvements over a GMM/HMM baseline system when the same features used in the baseline system are also used to generate AE-BN features. Taking the lessons learned on the 50-hour task, we then explore AE-BN features on a larger 430-hour Broadcast News task, where we observe that the AE-BN features offer a 0.5% improvement over a strong GMM/HMM baseline with a WER of 16.0%. Finally, system combination of the AE-BN and baseline systems provides an additional 0.5% absolute improvement over the AE-BN system alone, giving a final WER of 15.0%.

The rest of this paper is organized as follows. Section 2 describes the AE-BN system. Section 3 summarizes the experiments performed, while the analysis of AE-BN features on 50-hours of Broadcast News is presented in Section 4. Results using AE-BN features on 430-hour of Broadcast News is presented in Section 5 while system combination results are discussed in Section 6. Finally, Section 7 concludes the paper and discusses future work.

2. BOTTLENECK AUTO-ENCODER

2.1. Feature Extraction

A diagram of our bottleneck auto-encoder (AE-BN) system is depicted in Figure 1. First, given a set of input features, a DBN is pre-trained and then fine-tuned using backpropagation to minimize the cross-entropy between the set of target and hypothesized class probabilities. In this DBN architecture, the user specifies number of layers, number of hidden units per layer (i.e., 1024) and number of output targets (i.e., 384). This first step is similar to DBN training done for speech recognition applications [6], [7].

After DBN training, a neural network auto-encoder (AE) with a BN layer of 40 is trained to reduce the dimensionality of the output targets. The input to the AE is the 384 unnormalized log-posterior probabilities taken before the softmax output layer. We use two layers to reduce 384 output targets to 40, where each layer reduces the dimensionality of the previous layer by roughly a factor of three. A softsign nonlinearity ($y = x/(1 + |x|)$) is used between layers, which has been shown to be effective when training DBNs [10]. The training criterion for the AE is the cross-entropy between the normalized posteriors produced by processing the AE input and output through a softmax. Once the AE is trained, we extract features using the DBN weights and the weights of the AE up to the 40-dimensional bottleneck before the softsign nonlinearity. As in [3], an LDA is applied to these features and then a GMM/HMM acoustic model is built from these features.

2.2. Acoustic Model Training

A typical state-of-the-art LVCSR system [7] utilizes a specific recipe during acoustic model training which makes use of feature-space speaker adaptation (FSA), including vocal tract length normalization (VTLN) and feature space Maximum Likelihood Linear Regression (fMLLR), followed by discriminative training (DT). Each additional stage in this recipe typically uses more powerful modeling techniques. Bottleneck features are a type of frame-level discriminative feature when the cross-entropy training criterion is used to train the DBN [11]. However, discriminative training of GMM/HMM systems can be thought of as a sequence-level discriminative technique, since typically this objective function is created from a set of correct and competing hypotheses of the training data. Since speech recognition is a sequence-level problem, usually sequence-level discriminative methods have been shown to be more powerful than frame-level discriminative methods [11].

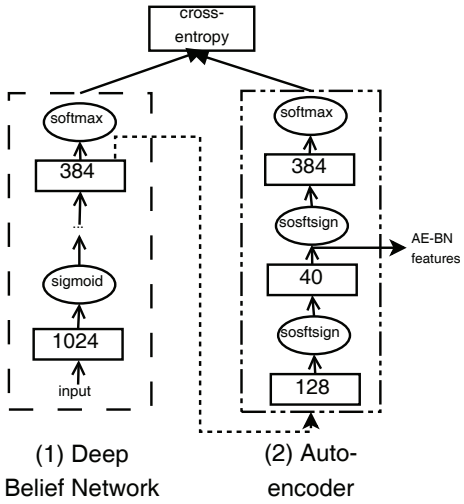

Fig. 1. Structure of DBN and Bottleneck Auto-Encoder. The dotted boxes indicate modules that are trained separately.

FSA move speech features into a canonical feature space. We hypothesize that extracting AE-BN features before FSA and then subsequently applying FSA would undo some of the frame-level discrimination in the AE-BN features. Similarly, if AE-BN features are created after fBMMI, then some of the sequence-level discrimination might be undone. With this intuition, we decide to create our AE-BN features after the FSA stage, where we still obtain the benefits of a canonical feature space without undoing any sequence-level discrimination. After AE-BN features are extracted and a GMM/HMM system is trained via maximum-likelihood on these features, we then apply feature and model-space DT. In Section 4.2, we show experiments to support our intuition of creating AE-BN features after FSA.

2.3. System Combination

BN features derived from NNs are usually complementary to baseline systems built from typical short-time speech features. Therefore, combining BN and baseline systems, either through tandem [3] or model-combination [10], is typically done to improve system performance. Even though our AE-BN features are extracted from a DBN built using short-time speech features, we hypothesize that the deepness of the DBN transforms the original speech features into a new space which could be complementary to the original features. In this paper, we explore model-combination, a system combination approach where the acoustic scores are computed as a weighted linear combination of scores from the two or more systems that can have different decision trees.

3. EXPERIMENTS

3.1. Corpora

Our experiments are conducted on an English Broadcast News transcription task [11]. Two different acoustic models are used which are

trained on 50 hours and 430 hours of data from the 1996 and 1997 English Broadcast News Speech collections and English broadcast audio from TDT-4. The initial acoustic features are 19-dimensional PLP features. Feature-space speaker adaptation (FSA), including VTLN and fMLLR is first performed. Next, a set of discriminatively trained features and models are created using the boosted Maximum Mutual Information (BMMI) criterion. Finally, models are adapted via MLLR to produce a state-of-the-art baseline GMM/HMM system. The acoustic models trained on 50 hours have 3,000 states and 50,000 Gaussians (4.1M trainable parameters), while the acoustic models trained on 430 hours have 6,000 states and 150,000 Gaussians (12.2M trainable parameters). Results are reported on the EARS Dev-04f set.

3.2. DBN+AE-BN Training

Unless otherwise specified, all DBNs use FSA features as input. In [7], it was observed that a 6-layer DBN with 1,024 hidden units per layer was an appropriate architecture for Broadcast News tasks. All DBNs are pre-trained using the procedure outlined in [7]. During fine-tuning, the final output layer is a softmax nonlinearity with 384 output targets. Unless otherwise noted, we use 384 output targets in all DBN experiments, obtained by clustering the context-dependent states in the baseline GMM/HMM system. After one pass through the data, loss is measured on a held-out set[2] and the learning rate is annealed (i.e. reduced) by a factor of 2 if the held-out loss has grown from the previous iteration [1]. Training stops after we have annealed the weights 5 times. Unless otherwise noted, all DBN results are reported using the cross-entropy loss function, due to computational benefits. Experiments with the sequence loss function [11] utilize the MPE objective function.

One the DBN is trained, the set of weights which provides the highest frame-classification accuracy on a held out set is taken as the final set of weights for generating the 384 output target log-probabilities, taken before the softmax layer. We use an AE architecture of 384-128-40-384 to reduce the dimensionality of the output targets. Again, after one pass through the data, the cross-entropy loss is measured on a held-out set and the learning rate is annealed by a factor of two if the held-out loss has increased from the previous iteration [1]. Training stops after we have annealed 5 times. The total number of trainable parameters in our DBN + AE system is 5.0M. Once the DBN and AE-BN are trained, AE-BN features are extracted. To fairly compare the performance of the AE-BN features to the baseline GMM/HMM system, the acoustic model on the AE-BN features is trained with the same number of states and Gaussians as the baseline system.

4. ANALYSIS OF AE-BN FEATURES

4.1. Impact of Improving DBN

Past research has shown that DBN performance is improved with deeper networks, pre-training and using a better training criterion [6], [7]. In this section, we study the impact that a better DBN has on the AE-BN features. First, rows (a) and (b) in Table 1 show the WER of the AE-BN features when the DBN was trained with 4 versus 6 layers. The table confirms that a deeper network improves the AE-BN features, similar to the trend reported in [8]. Second, the effect of pre-training on the AE-BN features can be compared in row (b) which uses a pre-trainined DBN and row (c) which uses a randomly initialized MLP. The results indicate that pre-training also improves the AE-BN system. Finally, row (d) illustrates that using the sequence-level training criterion provides improvements to the AE-BN system compared to using the cross entropy criterion in row (b).[3]

Architecture	AE-BN WER
(a) DBN - 4 Layers	22.5
(b) DBN - 6 Layers	21.5
(c) MLP - 6 Layers	22.2
(d) DBN - Seq.	20.6

Table 1. Effect of Deeper Network

4.2. AE-BN Performance With Different Features

In this section, we explore what is the best feature space to extract AE-BN features from. The following trends can be observed. First, row (a) in Table 2 indicates a WER of 22.6% when the DBN and AE are trained using VTLN features. Applying fMLLR processing to AE-BN, the WER is 21.8%. One can argue that applying fMLLR transforms AE-BN features to a canconical space and can undo some of the frame-level discrimination of the features. This can be justified more clearly by row (b), which indicates that if the DBN and AE are trained on VTLN+fMLLR features which preserves the canconical feature space, then the AE-BN is at 21.5%, which is slightly better than 21.8%. Finally, row (c) indicates that training the DBN and AE on fBMMI features results in a WER of 21.5%, which is worse than training the DBN on VTLN+fMLLR features, and then applying discriminative training. Again, if AE-BN features are extracted after the sequence-level discriminative fBMMI features, some of the sequence-level power can be lost. This justifies our intuition that AE-BN features should be extracted after the FSA stage in the LVCSR recipe, to appropriately capture its frame-level discriminative power.

Feature	AE-BN WER
(a) VTLN	22.6 +fMLLR: 21.8
(b) +fMLLR	21.5 +fBMMI: 19.3
(c) +fBMMI	21.5

Table 2. Effect of Different Features on AE-BN

4.3. Comparison of AE-BN to Other Features and Methods

In this section, we study the performance of the AE-BN features after feature and model space DT is performed. Table 3 shows the results for two AE-BN feature sets, one where the DBN is trained using the cross-entropy criterion and the other using the sequence criterion. We also compare to a baseline GMM/HMM system as well as a strong DBN hybrid system first described in [7]. This DBN is a 6-layer DBN with 1,024 hidden units and output targets equal to the number of context-dependent states of the HMM (2,220). The DBN is trained with fBMMI features using the sequence criterion.

The table indicates that AE-BN features trained via the sequence criterion offers the best performance of all methods. The AE-BN features provide a 0.9% absolute improvement over the hybrid DBN system, which was not observed in [8]. We hypothesize that using an

[2]Note that this held out set is different than Dev-04f.

[3]The authors are aware that having increased number of output targets also improves DBN performance [7]. The impact of increased output targets on the AE-BN will be explored in the future.

AE-BN structure as opposed to a regular bottleneck structure allows for frame accuracy to be preserved at the output targets, allowing for the AE-BN features to outperform the hybrid system. The 17.5% WER is the best result to date on the Dev-04f task, using an acoustic model trained on 50 hours of data [12].

Feature Space	GMM/HMM Baseline	AE-BN Cross Ent.	AE-BN Seq.	DBN Hybrid
FSA	24.8	21.5	20.6	
+fBMMI	20.7	19.3	19.0	
+BMMI	19.6	18.4	18.1	
+ MLLR	18.8	18.0	**17.5**	18.4

Table 3. WER, Models Trained on 50 Hours

Since AE-BN can be thought of as a type of frame-level discriminative training, to justify the result further we explored the behavior of the baseline system when two rounds of discriminative training are performed. Specifically, once an original fMMI transform is learned, new lattices are created and another round of feature and model-space discriminative-training is performed. Double fBMMI+BMMI+MLLR provides a WER of 18.4% which is worse than both AE-BN features, further demonstrating the benefit of the AE-BN features.

5. RESULTS ON A LARGER TASK

Now that we have studied the behavior on a small 50 hour BN task, in this section we explore the behavior of the AE-BN features on larger 430 hour Broadcast News task. Given FSA features (VTLN+fMLLR), we train a 6 layer DBN 384 output targets, the same architecture as the DBN trained on 50 hours. Note that because sequence training is computationally expensive, we only consider the cross-entropy criterion for these experiments. After DBN training, we then train an AE given these 384 output probabilities to extract bottleneck features, and perform maximum-likelihood GMM/HMM training. Next, we perform feature and model-space discriminative training to both the baseline FSA and AE-BN features. Table 4 shows results at different stages of the LVCSR recipe. Even on the larger task, the AE-BN offers a 0.5% absolute improvement over the baseline GMM/HMM system.

Feature Space	Baseline	AE-BN
FSA	20.2	17.6
+fBMMI	17.7	16.6
+BMMI	16.5	15.8
+MLLR	16.0	**15.5**

Table 4. WER, Models Trained on 430 Hours

6. MODEL COMBINATION

Finally, we explore the complementarity of the AE-BN and baseline methods by performing model combination on both the 50 and 430 hour tasks. Table 5 shows that model-combination provides a 1.1% absolute improvement over individual systems on the 50 hour task, and a 0.5% absolute improvement over the individual systems on the 430 hour task, confirming the complementarity of the AE-BN and baseline systems.

Method	50 Hours	430 Hours
(a) Baseline	18.8	16.0
(b) AE-BN	17.5	15.5
(c) Model Combination (a)+(b)	**16.4**	**15.0**

Table 5. Results Using Model Combination

7. CONCLUSIONS

In this paper, we explored pre-trained, deeper networks to extract auto-encoder bottleneck (AE-BN) features from a DBN. On a 50 hour task, we showed that the AE-BN features offered more than a 0.9% absolute improvement over strong DBN and GMM/HMM systems. In addition, on a larger 430 hour task, the AE-BN features provided a 0.5% absolute improvement over the GMM/HMM baseline. To our knowledge, this is the first use of BN features offer improvements over a GMM/HMM baseline when the same features used in the baseline system are also used to generate AE-BN features [3]. In addition, to our knowledge this is the first use of BN features to offer improvements over a pre-trained, deep DBN with output targets equal to the number of context-dependent states [8].

8. ACKNOWLEDGEMENTS

The authors would like to thank Hagen Soltau, George Saon and Stanley Chen for their contributions towards the IBM toolkit and recognizer utilized in this paper. In addition, thank you to Petr Fousek, Petr Novak, Christian Plahl and Abdel-rahman Mohamed for useful discussions related to DBNs and bottleneck features.

9. REFERENCES

[1] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, 1993.

[2] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem Connectionist Feature Extraction for Conventional HMM Systems," in *Proc. ICASSP*, 2000.

[3] F. Grezl, M. Karafiat, S. Kontar, and J. Cernocky, "Probabilistic and Bottleneck Features for LVCSR of Meetings," in *Proc. ICASSP*, 2007.

[4] A. Mohamed, G. E. Dahl, and G. E. Hinton, "Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition," *IEEE TSAP*, 2011.

[5] G. E. Hinton, S. Osindero, and Y. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[6] F. Seide, G. Li, and D. Yu, "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks," in *Proc. Interspeech*, 2011.

[7] T. N. Sainath et. al., "Making Deep Belief Networks Effective for Large Vocabulary Continuous Speech Recognition," in *to appear in Proc. ASRU*, 2011.

[8] D. Yu and M. L. Seltzer, "Improved Bottleneck Features Using Pre-trained Deep Neural Networks," in *Proc. Interspeech*, 2011.

[9] F. Grezl and P. Fousek, "Optimizing Bottleneck Features for LVCSR," in *Proc. ICASSP*, 2008.

[10] L. Mangu et. al., "The IBM 2011 GALE Arabic Speech Transcription System," in *to appear in Proc. ASRU*, 2011.

[11] B. Kingsbury, "Lattice-Based Optimization of Sequence Classification Criteria for Neural-Network Acoustic Modeling," in *Proc. ICASSP*, 2009.

[12] G. Saon and J. T. Chien, "Discriminative Training for Bayesian Sensing Hidden Markov Modlels," in *Proc. ICASSP*, 2011.