From Undercomplete to Sparse Overcomplete Autoencoders to Improve LF-MMI Speech Recognition

Selen Hande Kabil[1,2]*, Hervé Bourlard*[1,2]

[1]Idiap Research Institute, Martigny, Switzerland
[2]École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland
{selen.kabil, bourlard}@idiap.ch

Abstract

Starting from a strong Lattice-Free Maximum Mutual Information (LF-MMI) baseline system, we explore different autoencoder configurations to enhance Mel-Frequency Cepstral Coefficients (MFCC) features. Autoencoders are expected to generate new MFCC features that can be used in our LF-MMI based baseline system (with or without retraining) towards speech recognition improvements. Starting from shallow undercomplete autoencoders, and their known equivalence with Principal Component Analysis (PCA), we go to deeper or sparser architectures. In the spirit of kernel-based learning methods, we explore alternatives where the autoencoder first goes overcomplete (i.e., expand the representation space) in a nonlinear way, and then we restrict the autoencoder by means of a sequent bottleneck layer. Finally, as a third solution, we use sparse overcomplete autoencoders where a sparsity constraint is imposed on the higher-dimensional encoding layer. Experimental results are provided on the Augmented Multiparty Interaction (AMI) dataset, where we show that all aforementioned architectures improve speech recognition performance, although with a clear advantage on sparse overcomplete autoencoders for both close-talk and far-field speech sets.

Index Terms: pca, bottleneck, sparse overcomplete autoencoder, chain models, speech recognition

1. Introduction

Automatic Speech Recognition (ASR) is the process of converting the speech signal into its corresponding sequence of words or other linguistic components. With the advances in big data and computing power, ASR technologies have inevitably evolved and adopted for many applications. For instance, consumer-centric applications which involve voice search and interactions with mobile devices and home entertainment systems have emerged. These everyday life applications require ASR systems to be robust to full-range of real-world noise and other acoustic distortions. Therefore, maintaining robustness remains to be an important research direction in ASR field.

To reach noise robust ASR, a number of techniques have been proposed. These methods mainly differ based on their focus on different components in the speech recognition pipeline. For instance, the main goal of the method can be enhancing the speech feature extraction stage or robust modeling of the recognizer. Techniques like Linear Discriminant Analysis (LDA) [1], Heteroscedastic Linear Discriminant Analysis (HLDA) [2] have been proposed to improve the discriminating capabilities of the original features. Similarly, Principal Component Analysis (PCA) [3, 4] and Kernel PCA [5, 6] have been found useful for producing features with better recognition performance. With the advances in deep learning for ASR, speech feature enhancement by means of Denoising Autoencoders (DAE) has been widely investigated [7, 8, 9, 10, 11]. Thanks to their ease of use, the enhanced denoised features from DAEs can then be propagated to the back-end acoustic model [12].

In this paper, we aim to present a comparative study on the use of autoencoders for robust speech recognition with LF-MMI systems. We examine and compare the potential of different autoencoders for producing new MFCC features with better recognition performance. Starting from shallow undercomplete autoencoders, and their known equivalence with PCA, we go to deeper or sparser architectures. In the spirit of kernel-based learning methods, we explore alternatives where the autoencoder representation space is first expanded in a nonlinear way and then restricted by means of a sequent bottleneck layer. Finally, as a third solution, we use sparse overcomplete autoencoders where a sparsity constraint (based on ℓ_1 norm penalty) is imposed on the higher-dimensional encoding layer.

Our experimental results on Augmented Multiparty Interaction (AMI) dataset show that all architectures improve speech recognition perfrormance. However, sparse overcomplete autoencoder, where a sparsity constraint (based on ℓ_1 norm minimisation) is imposed on the higher-dimensional encoding layer, is the best performer on both close-talk and far-field speech. These findings underlines that our new MFCC features are indeed better features compared to their original counterparts for discriminating speech classes.

This paper is organized as follows; after we elaborate on the background information about autoencoders in Section 2, we present the proposed approach and baseline system in Section 3, the experimental setup and results in Section 4, and finally the conclusion in Section 5.

2. Autoencoders

Autoencoder (AE) [13] is a neural network whose goal is to reconstruct d-dimensional input feature vectors as d-dimensional output vectors. An AE consists of two main components: encoder and decoder. Encoder produces the code (i.e., encoding, embedding) given the input. Decoder takes this code and tries to reconstruct the original input at the output layer. AE generally aims to minimize Mean Square Error (MSE) between the original input and reconstructions.

Based on the model configurations, AEs can be categorized. For instance, if the encoding layer has lower dimensionality than the input, the model is called *undercomplete autoencoder*. However, if the encoding layer has higher dimensionality, then the model is called *overcomplete autoencoder*. If the model has only one hidden layer, it is called *shallow autoencoder*. Oppositely, if the model has more than one hidden layers (while still preserving the symmetry of the network), it is called *deep autoencoder*.

Since the objective of AE is to reconstruct its input, for the sake of perfect reconstruction, it can simply copy the input to the output layer instead of learning meaningful encodings. This undesired phenomenon (i.e., identity mapping) happens when the modeling capacity of AE is high. Here, by modeling capacity, we mean the complexity of the relationships in the data (i.e., patterns) that the model can express. A rough estimate for the capacity of a model can be made by simply counting the number of its parameters. More parameters indicate higher capacity. In other words, given the data, if the model tends to overfit thanks to its high modeling capacity, it learns identity mapping.

In the case of identity mapping, different constraints can be enforced on the model to regularize its modeling capacity. These constraints can be in the form of restricting the encoding layer dimension of the model (e.g. undercomplete autoencoders). Also, additional regularization term(s) can be introduced to the AE loss function to force AE to learn meaningful encodings (e.g. sparse overcomplete autoencoders).

In the following subsections, we provide background information about the autoencoder configurations that we used in our experiments.

2.1. Undercomplete autoencoder

Shallow undercomplete autoencoder learns to span the same subspace as PCA under certain conditions such as linear decoder, MSE as loss function and real-valued input data, as shown in [14]. Having smaller code dimension than the input dimension (depicted as $d >> p$ in Figure 1) forces the autoencoder to learn the salient features in the training data.

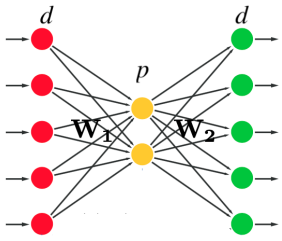

Figure 1: *Shallow undercomplete autoencoder where the code dimension p is smaller than the input dimension d.*

2.2. Deep undercomplete autoencoder with nonlinear space expansion

As the composition of linear operations yields another linear operation at the end, stacking linear layers for the sake of building a deep autoencoder is actually pointless. Hence, for our research, in the spirit of Support Vector Machines (SVM) and other kernel-based learning methods [15, 16], we first expand the space (depicted as $d << q$ in Fig. 2) in a nonlinear fashion so that the input features are projected into a high-dimensional space where the relational latent factors in the input are easier to model. Then, we apply compression with a bottleneck layer ($d >> p$) so that we can extract the salient features of the projected input data.

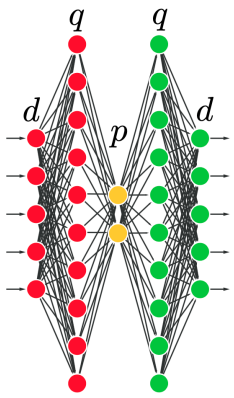

Figure 2: *Deep undercomplete autoencoder with space expansion where q and p stand for the expanded space dimension and the the bottleneck code dimension respectively.*

2.3. Sparse overcomplete autoencoder

Unless some constraint is applied on the modeling capacity, the shallow overcomplete autoencoder can simply learn the identity mapping (i.e., copying inputs to outputs for perfect reconstruction). To avoid this, sparsity is enforced by constraining the hidden unit activations (i.e., encodings). The sparsity constraints is in the form of additional penalty term (i.e., regularizer term) in the loss function. The hyperparameter λ weights the penalty term for learning meaningful representations, with respect to the reconstruction loss.

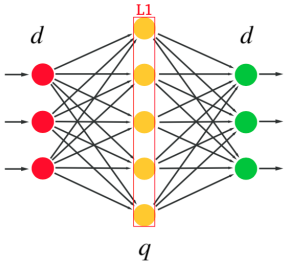

Figure 3: *Shallow sparse overcomplete autoencoder with ℓ_1 norm penalty enforced on encodings (i.e., encoding layer activations).*

Sparse autoencoders with ℓ_1 norm penalty on encodings (Figure 3) seems to be the most natural choice, and is also used in sparse coding [17]. Hence, we use this configuration for our experiments.

3. Proposed Approach and Baseline System

3.1. Proposed approach

In this paper, we aim to reinforce our understanding of autoencoders in the context of speech feature enhancement for LF-MMI system. Despite their known limitations, Mel-Frequency Cepstral Coefficients (MFCC) are often in use due to their low correlation and their compact, computationally efficient nature.

Following our proposed approach (Fig. 4), we feed new MFCC features (i.e., autoencoder reconstructions) to the LF-MMI acoustic model. Therefore, we examine the potential of a

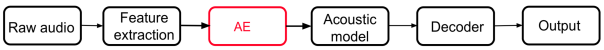

Figure 4: *Rough illustration for our proposed approach. Instead of the original MFCCs, the reconstructions from the autoencoder is fed to the acoustic model. The autoencoders used in experiments are introduced in Section 2. Further details about the acoustic model can be examined in Fig. 5.*

group of autoencoders for producing better MFCC features for robust recognition.

In addition, we investigate the impact of using the new MFCCs with three scenarios: (1) only for decoding (i.e., keeping the acoustic model parameters frozen), (2) adapting the acoustic model on the new features and (3) training a new acoustic model from scratch with the new features. Adaptation of the acoustic model does not yield performance improvement for any of the configurations in Section 2. Therefore, we present the results for the only decoding and training from scratch scenarios in Section 4.

3.2. Baseline system

The experiments are conducted on AMI corpus [18] which contains recordings of spontaneous conversations in meeting scenarios in English. The corpus provides audio recordings from close-talk (stated as IHM) and far-field (stated as SDM) microphones. Both close-talk and far-field speech streams have been recorded in parallel. The dataset is available at 16 kHz sampling rate with nearly 100 hours of meeting recordings divided approximately as 81 hours train set, 9 hours development and 9 hours evaluation set.

Table 1: *The recognition performance (in WER%) for the baseline LF-MMI systems on close-talk IHM and far-field SDM evaluation sets.*

Architecture	IHM	SDM
LF-MMI acoustic model	19.7	41.7

The configuration for the IHM acoustic model is presented in Fig. 5. The model configuration for SDM is same, except it does not contain the CNN and LSTM layers.

Two acoustic models are trained using IHM and SDM dataset with the LF-MMI criteria [19] following the standard chain model recipe in Kaldi speech recognition toolkit [20]. The input is high resolution MFCC features with d =40. The output of the systems is the pseudo-log-likelihoods with dimension of 176. The Word Error Rate (WER) for IHM and SDM are 19.7% and 41.7% respectively, as shown in Table 1.

4. Experimental Setup and Results

4.1. Undercomplete autoencoder

The undercomplete autoencoder takes MFCC features with d = 40 as input, encodes it into compact, low-rank encodings and then outputs the reconstructions as new MFCC features to be use in the rest of the speech recognition pipeline as shown in Figure 4. The low-rank encoding dimension p is 30. The autoencoders are implemented in Pytorch [21] and trained with MSE using stochastic gradient descent with batch size 256 and learning rate scheduler (initial learning rate set as 0.1).

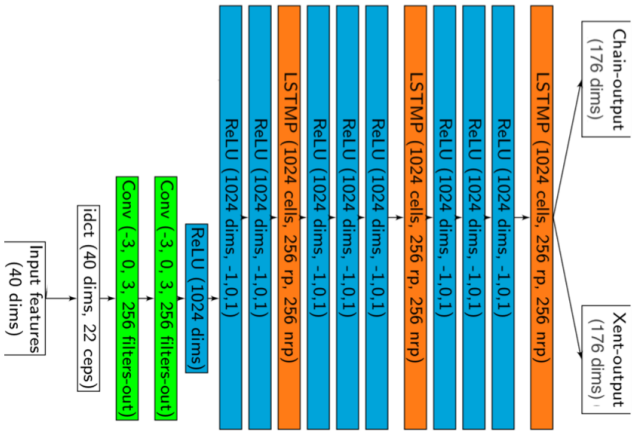

Figure 5: *The model configuration for IHM baseline. The green blocks represent CNN layers. The blue blocks represent TDNN layers with RELU activation function. The orange blocks denote LSTM layers. The xent-output layer is used for regularization purpose only.*

Table 2: *The recognition performance (in WER%) when the acoustic model parameters are kept fixed.*

	IHM AE	SDM AE
data= IHM	21.6	20.5
data= SDM	48.5	43.2

To clarify notation used in Table 2, it is worth to mention that data=IHM indicates that the acoustic model is previously trained on IHM and the streaming data for decoding task is also IHM (eval set). SDM AE indicates that AE is previously trained on SDM data. Hence, from Table 2, we can state the using SDM AE for projecting IHM data before passing to the acoustic model which is previously trained on IHM results in WER=20.5%.

Table 3: *The recognition performance (in WER%) when a new acoustic model is trained from scratch with the new MFCC features. Note that the same model configuration and training procedure for the baseline acoustic model is used for training the new LF-MMI based acoustic model from scratch.*

	IHM AE	SDM AE
data= IHM	19.4	19.3
data= SDM	41.9	41.2

Before commenting on the results, we want to clarify some of the notations used in Table 3. For instance, data=IHM denotes that the original streaming data is IHM. SDM AE indicates that the AE is previously trained on SDM data. Hence, we say that using new features (obtained by projecting the IHM MFCC original features with SDM AE) for training a new acoustic model improves the recognition performance for IHM (0.4% absolute improvement, from 19.7% to 19.3%). The column-wise comparison of the results in Table 3 shows that SDM AE has better generalization power, probably because far-field SDM data has more acoustic variation compared to close-talk IHM.

4.2. Deep undercomplete autoencoder with nonlinear space expansion

The deep undercomplete autoencoder takes MFCC feature vectors with dimension d=40 as input. First, it projects these features into a higher-dimensional space q=1760 in a nonlinear fashion so that the relational latent factors in the input are easier to model. Then, it applies compression with a bottleneck layer (p=30). Finally, it outputs the reconstructed MFCCs as new feature set. Except for the undercomplete (bottleneck) encoding layer and output layer, Sigmoid activation function is used for introducing nonlinearity to the model.

Table 4: *The recognition performance (in WER%) when acoustic model parameters are frozen.*

	IHM AE	SDM AE
data= IHM	51.0	50.2
data= SDM	70.0	79.7

Table 4 presents the recognition results while the acoustic model parameters are kept fixed. For both datasets, we observe serious degradation in the performance. The fact that the baseline system obtains 19.7% WER for data=IHM case (as shown in Table 1) shows that the LF-MMI based baseline acoustic model is indeed strong. However, the results in Table 4 for data=IHM display that baseline acoustic model is tuned finely to the original MFCC features, and hence sensitive to the changes in the input data.

Table 5: *The recognition performance (in WER%) when a new acoustic model is trained from scratch with the new MFCC features.*

	IHM AE	SDM AE
data= IHM	19.6	19.5
data= SDM	41.6	41.5

Table 5 presents the recognition results for training a new LF-MMI based acoustic model from scratch with new MFCC features. It is important that the same model configuration and training procedure for the baseline acoustic model is used for training the new model. For both data=IHM and data=SDM cases, we obtain improvements in WER, compared to the baseline system. This indicates that our proposed approach indeed extracts better features for recognition. The best results (19.5% for IHM and 41.5% for SDM) are obtained when off-the-shelf SDM AE is used for projecting the original MFCC features. This is due to the generalization power of the SDM AE. This behaviour is also observed for the undercomplete autoencoders as shown in in Table 3.

4.3. Sparse overcomplete autoencoder

The sparse overcomplete autoencoder takes MFCC feature vectors with dimension d=40 as input, encodes them into high-dimensional sparse encodings with dimension q=1760 and then outputs the reconstructions as new MFCC features to be use in the rest of the speech recognition pipeline as shown in Figure 4. The autoencoders are implemented in Pytorch [21] and trained with MSE using stochastic gradient descent with batch size 256 and learning rate scheduler (initial learning rate 0.1). In addition, as an early-stopping mechanism, autoencoder training is terminated when the loss on the development set is not improved for 10 consecutive epochs. For λ, grid search is performed on $[10^0, 10^{-6}]$. The model with the optimal λ is determined based on the WER on the development set.

Table 6: *The recognition performance only decoding (in WER%) when the acoustic model parameters are kept fixed.*

	IHM AE	SDM AE
data= IHM	23.3	22.5
data= SDM	42.3	46.4

In Table 6, similar to Table 4, we observe degradation stemming from the baseline acoustic model finely tuned to the original MFCC features.

It is important to note that all the results (data=IHM and data=SDM) in Table 7 are better than baseline performance, but also other systems with different autoencoder configurations. This highlights the importance of the presence of sparsity for robustness.

Table 7: *The recognition performance only decoding (in WER%) when a new acoustic model is trained from scratch with the new MFCC features. Note that the same model configuration and training procedure for the baseline acoustic model is used for training the new LF-MMI based acoustic model from scratch.*

	IHM AE	SDM AE
data= IHM	19.6	19.3
data= SDM	41.5	41.3

5. Conclusion

In this paper, we aim to explore the potential of different autoencoder configurations to improve MFCC features for LF-MMI based speech recognition system. Starting from shallow undercomplete autoencoders, and their known equivalence with PCA, we go to deeper or sparser architectures. In the spirit of kernel-based learning methods, we explore alternatives where the autoencoder first goes overcomplete (i.e., expand the representation space) in a nonlinear way and then restrict the autoencoder by means of a sequent bottleneck layer. Finally, as a third solution, we use sparse overcomplete autoencoders where a sparsity constraint (based on L1 norm minimisation) is imposed on the higher-dimensional encoding layer.

Our experiments on AMI dataset shows that when the new features are used only for decoding (i.e., keeping the baseline acoustic model parameters fixed), the performance degrades. This is due to the fact that baseline LF-MMI acoustic model is finely tuned to the original MFCC features. Similarly, when the baseline acoustic model is further trained on the new MFCC features, we again observe performance degradation. This hints that the original and new MFCC features have different data characteristics. And finally, when a new LF-MMI based acoustic model is trained from scratch on the new MFCC features, we observe improvements on WER.

6. Acknowledgements

This work was funded by the Swiss National Science Foundation under the project Sparse and HIerarchical Structures for Speech Modeling (SHISSM).

7. References

[1] X. Aubert, R. Haeb-Umbach, and H. Ney, "Continuous mixture densities and linear discriminant analysis for improved context-dependent acoustic models," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2. IEEE, 1993, pp. 648–651.

[2] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank hmms for improved speech recognition," *Speech communication*, vol. 26, no. 4, pp. 283–297, 1998.

[3] T. Takiguchi and Y. Ariki, "Pca-based speech enhancement for distorted speech recognition." *Journal of multimedia*, vol. 2, no. 5, 2007.

[4] S.-M. Lee, S.-H. Fang, J.-w. Hung, and L.-S. Lee, "Improved mfcc feature extraction by pca-optimized filter-bank for speech recognition," in *IEEE Workshop on Automatic Speech Recognition and Understanding, 2001. ASRU'01.* IEEE, 2001, pp. 49–52.

[5] C. Leitner, F. Pernkopf, and G. Kubin, "Kernel pca for speech enhancement," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[6] A. Lima, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura, "On the use of kernel pca for feature extraction in speech recognition," *Rn*, vol. 2, no. w1, p. w3.

[7] T. Ishii, H. Komiyama, T. Shinozaki, Y. Horiuchi, and S. Kuroiwa, "Reverberant speech recognition based on denoising autoencoder." in *Interspeech*, 2013, pp. 3512–3516.

[8] X. Feng, Y. Zhang, and J. Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 1759–1763.

[9] F. Weninger, S. Watanabe, Y. Tachioka, and B. Schuller, "Deep recurrent de-noising auto-encoder and blind de-reverberation for reverberated speech recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4623–4627.

[10] M. Mimura, S. Sakai, and T. Kawahara, "Reverberant speech recognition combining deep neural networks and deep autoencoders augmented with a phone-class feature," *EURASIP journal on Advances in Signal Processing*, vol. 2015, no. 1, pp. 1–13, 2015.

[11] T. Gao, J. Du, L. Dai, and C. Lee, "Joint training of front-end and back-end deep neural networks for robust speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4375–4379.

[12] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 14–22, 2011.

[13] M. Kramer, "Autoassociative neural networks," *Computers & chemical engineering*, vol. 16, no. 4, pp. 313–328, 1992.

[14] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological cybernetics*, vol. 59, no. 4, pp. 291–294, 1988.

[15] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 1999.

[16] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambrigde, 2000.

[17] B. A. Olshausen and D. J. Field, "Sparse coding of sensory inputs," *Current opinion in neurobiology*, vol. 14, no. 4, pp. 481–487, 2004.

[18] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The ami meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, 2005, p. 100.

[19] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," in *Proceedings of Interspeech*, 2016, pp. 2751–2755.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Proceedings of the IEEE 2011 workshop on automatic speech recognition and understanding*, 2011.

[21] A. Paszke, S. Gross, S. Chintala, and G. Chanan, "Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration," 2017.