Interspeech 2018
2-6 September 2018, Hyderabad

Speech enhancement using deep mixture of experts based on hard expectation maximization

Pavan Karjol and Prasanta Kumar Ghosh

Electrical Engineering, Indian Institute of Science, Bengaluru 560012, India

pavank@iisc.ac.in, prasantg@iisc.ac.in

Abstract

We consider the problem of deep mixture of experts based speech enhancement. The deep mixture of experts, where experts are considered as deep neural network (DNN), is difficult to train due to the network structure. In this work, we propose a pre-training method for individual DNN in deep mixture of experts. We use hard expectation maximization (EM) to pre-train the individual DNNs. After pre-training, we take a weighted combination of outputs of individual DNN experts and jointly train the whole system. We compare the proposed method with single DNN based speech enhancement scheme. Speech enhancement experiments, in four SNR conditions, show the superiority of the proposed method over the baseline scheme. The average improvements obtained for four seen noise cases over single DNN scheme are 0.08, 0.59 dB and 0.015 in terms of objective measures viz perceptual evaluation of speech quality (PESQ), segmental signal to noise ratio (seg SNR) and short time objective intelligibility (STOI) respectively.

Index Terms: Deep neural networks, Hard expectation maximization, Speech enhancement

1. Introduction

Speech enhancement has been an important field of research for several decades as it plays a prominent role in many applications such as speech communication, automatic recognition system, hearing aids etc. [1]. The goal of speech enhancement is to suppress the noise in a noisy speech recording keeping the speech distortion to a minimum level.

Neural networks have emerged as powerful tools for many artificial intelligence tasks providing promising results. Their ability to capture complex variations in the input data makes them attractive for many applications including speech enhancement. Initially shallow networks were trained [2–4] to directly estimate clean speech from noisy spectrum. However, the objective function of neural network is typically non convex and complex to optimize. Due to lack in sophisticated computational resources and optimization techniques, such networks did not provide satisfactory results. In recent years, various techniques have been proposed [5–7] for better and faster training of neural networks. These modifications resulted in significant improvement in many areas including speech recognition and image recognition. Following the technique proposed in [5], a restricted Boltzmann machine (RBM) [8] was trained to initialize a deep neural network (DNN) for speech enhancement. Such an approach resulted in significant improvements in speech enhancement under training (seen) noise cases.

As an extension, multiple DNNs have been used for speech enhancement. These techniques can be broadly categorized as multiple experts in a more general context. There are many speech enhancement techniques where multiple experts are used. For instance, a mixture maximum model was proposed by Amit et al. [9], which is based on broad phoneme classes. However, it requires prior enhanced Mel frequency cepstral coefficient (MFCC) vectors specific to each phoneme, which reduces its generalizability across different speakers and also with respect to the intra broad phoneme class variability. Chazan et al. [10] employed phoneme information based pre-training method where forty DNNs (one for each phoneme class) and a classifier network were used to estimate speech presence probability (SPP). A separate DNN was pre-trained for each phoneme class to predict SPP, following which all the DNNs are trained jointly to estimate the overall SPP. Such a system outperformed single DNN based system in many test cases.

Instead of relying on phoneme labels, in this work, we explore a different groups of acoustic units that can be learnt in a data-driven manner to achieve an improved speech enhancement performance. We note that Chazan et al. [11] discussed the possibility of using generalized expectation maximization (EM) algorithm (we refer this variant of EM algorithm as soft EM) to train multiple DNNs instead of using phoneme labels. They also discussed the problems associated with such an approach. Specifically, they mention training complexity and convergence problems. In this work we try to solve those problems with a variant of EM algorithm viz hard EM. We propose to use task specific one epoch hard EM based pre-training of the multiple DNN system. In the scope of current work, the specific task refers to estimating clean spectrum from noisy one. The maximization step in each EM iteration (corresponding to parameters of DNN) is run just for one epoch. After each epoch, the data is redistributed among different DNNs in order to facilitate better estimation of clean speech. Following the work in [10], we combine these pre-trained DNNs, and train jointly. The proposed modification is found to reduce the training complexity, also results in simpler objective function. In addition, the proposed technique is found to outperform the baseline schemes viz single DNN, deep mixture of experts (dMoE) with soft EM based pre-training, dMoE without pre-training.

We use log spectrum as the target output instead of SPP in the proposed work. In the SPP based method, the noisy spectrum is assumed to be the sum of clean and noise spectra [10] which may not hold good at low SNRs. We assume that direct noisy spectrum to clean spectrum mapping could overcome this limitation. We conduct experiments using TIMIT (for clean speech utterances) [12] and Aurora (for noise signals) [13] databases. The number of noise types used for training is four. For testing, five unseen noise types are used in addition to these four seen noises. We consider four SNR conditions. Following the work of [9], we, in this work, use objective evaluation measures instead of any subjective assessment. We evaluate the proposed algorithm in terms of objective measures viz perceptual evaluation of speech quality (PESQ) [14], segmental signal

Authors thank Pratiksha trust for their support.

10.21437/Interspeech.2018-1730

to noise ratio (seg SNR) and short time objective intelligibility (STOI) [15]. We observe that the proposed method performs better than the baseline schemes in most of the cases especially for seen noise types at every SNR considered and unseen noise cases at high SNRs.

2. Deep mixture of experts

Deep mixture of experts (dMoE) is a special case of mixture of experts where the experts employed are DNNs. As shown in Fig. 1, the output of such a system $\hat{y}$ is given by,

$$\hat{y} = \sum_{q=1}^{N} p_q(x) f_q(x), \tag{1}$$

where x is an input to the system, $f_q(x)$ is the output of the q^{th} expert and $p_q(x)$ is the corresponding weightage given by the gating network, N is the number of experts (DNNs). In the scope of current work, $x, \hat{y}$ correspond to the log spectra of noisy speech and predicted clean speech respectively. When such a system is trained with input and target data, we expect to reduce the error ($\mathcal{E}$) between the predicted and original one, i.e., the following objective function has to be minimized.

$$\mathcal{E} = \frac{1}{M} \sum_{i=1}^{M} d\Big(y_i, \sum_{q=1}^{N} p_q(x_i) f_q(x_i)\Big), \tag{2}$$

where y_i is i^{th} clean speech log spectrum, x_i is the corresponding noisy one, M is the total number of training examples considered and $d()$ is an error metric like mean square error or mean absolute error etc. However, it has been noted that employing such an objective function may not converge sometimes especially as the complexity of the system increases [10]. Hence, Chazan et al. proposed to pre-train the individual DNNs [10] using phoneme information. In [11], the possibility of pre-training without using phoneme information has been discussed. Although it is not defined as pre-training, we, in the following subsection, show that it can be used as pre-training with required modifications and use these pre-trained DNNs to jointly train the whole system to minimize the objective function in eq. 2.

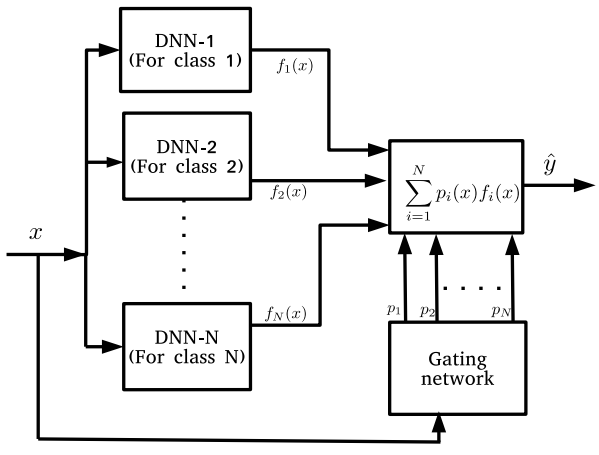

Figure 1: *Block diagram of dMoE*

2.1. Training using soft EM:

Chazan et al. [11] discussed the possibility of using soft EM to train dMoE. The problem considered involved estimation of probability values viz SPP. In this work, we consider estimation of log spectrum of clean speech unlike probability value. Hence, the analysis of training dMoE using soft EM presented here is slightly different. The log likelihood function to be maximized for such a system is given by,

$$\mathcal{L}(\theta) = \sum_{i=1}^{M} \log p(y_i|x_i;\theta) = \sum_{i=1}^{M} \log \sum_{q=1}^{N} p(y_i, q|x_i;\theta), \tag{3}$$

where q is the latent variable (the underlying DNN expert) and θ are the parameters of the overall system. The parameters can be decomposed as, $\theta = [\theta_1, \theta_2 \ldots \theta_N, \theta_c]$, where θ_q (with $q \in \{1, 2, \ldots, N\}$) are the parameters of q^{th} DNN expert and θ_c are the parameters of the gating network. For the considered problem i.e., the estimation of log spectrum of clean speech, we define (for notational simplicity we drop the subscript index i for further analysis and consider single example case. The same analysis can be extended to M number of training examples),

$$p(y|x, q;\theta_q) \propto \lambda \exp\Big(-\lambda \parallel y - f_q(x;\theta_q) \parallel^2 \Big), \tag{4}$$

where $f_q(x;\theta_q)$ is the output of the q^{th} DNN, and λ is the decay parameter which is fixed. The likelihood in eq. 3 can be maximized (with respect to θ) through EM algorithm (also known as soft EM). The expectation step Q at iteration $t+1$ given the parameters of the iteration t i.e., θ^t is as follows,

$$\mathcal{Q}(\theta;\theta^t) \propto \mathbb{E}_{p(q|x,y;\theta^t)}\Big[\log p(y|x, q;\theta_q) + \log p(q|x;\theta_c) \Big] \tag{5}$$

Considering all M training examples, the first term in eq. 5 can be written as (termed as Q_I),

$$Q_I \propto -\sum_{i=1}^{M} \sum_{q=1}^{N} p(q|x_i, y_i;\theta^t) \parallel y_i - f_q(x_i;\theta_q) \parallel^2, \tag{6}$$

Note that we expanded the expectation and substituted for $p(y_i|x_i, q;\theta_q)$ from eq. 4. Even though the analysis presented here is slightly different from that in [11], the problems associated with soft EM approach as discussed in [11] remain here as well – there is no closed form expression to perform maximization task. In addition, it may get stuck in local maxima if we simply maximize using stochastic gradient ascent [11]. Such maximization task is also computationally expensive [11]. However, we propose to use just one epoch training in the maximization step. We hypothesize that such modification can overcome the limitation discussed in [11]. We have to note that modified algorithm will still converge since increasing the auxiliary function is sufficient for the convergence of the EM algorithm [16]. However, with such a modification, soft EM based pre-trained dMoE doesn't provide really good results compared to single DNN approach. This could be due to the complicated objective function, i.e., as seen in eq. 6 the error on each example (x_i, y_i) is weighted by $p(q|x_i, y_i;\theta^t)$. In addition, even though we are training for one epoch, the training complexity is still high because each DNN has to be trained with entire data during each epoch.

2.2. Proposed method:

As discussed in previous section, the soft EM suffers from inherent problems when applied to training multiple DNN systems. To solve these problems, we propose not to use the weights, i.e., $p(q|x_i, y_i;\theta^t)$ as in eq (6), to weigh each sample,

rather aim to optimally select the expert DNN for each training sample separately (referred to as hard labeling of each sample with expert DNN index). We observe that such a setting automatically comes from starting with a different objective function given by (which, in literature, is known as the hard expectation maximization - hard EM [17]),

$$\theta^* = \arg\max_{\theta} \max_{\mathbf{q}} \sum_{i=1}^{M} \log p(y_i, q_i | x_i; \theta) \tag{7}$$

where $\mathbf{q} = [q_1, q_2, \ldots q_M]$ with $q_i \in \{1, 2, \ldots N\}$ and

$$p(y_i, q_i | x_i; \theta) = p(y_i | x_i, q_i; \theta_{q_i}) p(q_i | x_i; \theta_c) \tag{8}$$

In most of the analysis presented henceforth, q_i and q (which is not same as $\mathbf{q}$) can be interchanged (since it doesn't affect) except when finding the optimum latent variable for each data point, in which case the variable used is q_i like in eq. 7. The goal of the hard EM algorithm is to find optimum values of θ and $\mathbf{q}$ which maximizes eq. 7. The maximization task can be done by co-ordinate ascent method. In co-ordinate ascent we alternately optimize each variable keeping remaining variables fixed. The steps of the hard EM typically uses a co-ordinate ascent algorithm [17]. These are summarized below. Note that these steps are adapted from [17] with required modification for dMoE. The algorithm is as follows,

Initialize the parameters θ^0 and repeat the following steps until RHS of eq. 7 converges.

(I) Find the optimum latent variables $\mathbf{q}^{t+1} = [q_1^{t+1}, q_2^{t+1}, \ldots q_M^{t+1}]$ at iteration $t+1$ given θ^t as follows,

$$\mathbf{q}^{t+1} = \arg\max_{\mathbf{q}} \sum_{i=1}^{M} \log p(y_i, q_i | x_i; \theta^t), \tag{9}$$

which can be decoupled as,

$$q_i^{t+1} = \arg \max_{q_i \in \{1,2,\ldots N\}} p(y_i, q_i | x_i; \theta^t), \;\; 1 \leq i \leq M \tag{10}$$

(II) Find the optimum parameters $\theta^{t+1} = [\theta_1^{t+1}, \theta_2^{t+1}, \ldots, \theta_N^{t+1}, \theta_c^{t+1}]$ given $\mathbf{q}^{t+1}$ as follows,

$$\theta^{t+1} = \arg\max_{\theta} \sum_{i=1}^{M} \log p(y_i, q_i^{t+1} | x_i; \theta), \tag{11}$$

which can be decoupled as training of individual DNN experts and classifier DNN separately, i.e., $\forall q \in \{1, 2, \ldots, N\}$,

$$\theta_q^{t+1} = \arg\max_{\theta_q} \sum_{i: q_i^{t+1} = q} \log p(y_i | x_i, q; \theta_q) \tag{12}$$

$$= \arg\min_{\theta_q} \sum_{i: q_i^{t+1} = q} \| \; y_i - f_q(x_i; \theta_q) \; \|^2, \tag{13}$$

The eq. 13 is obtained by substituting eq . 4 in eq. 12 for each example considered. Note that Eq. 13 corresponds to training of individual DNN experts separately with data points that have been assigned from eq. 10 (i.e., from step **(I)**). The classifier training is as follows,

$$\theta_c^{t+1} = \arg\max_{\theta_c} \sum_{i=1}^{M} p(q_i^{t+1} | x_i; \theta_c) \tag{14}$$

Since the maximum value possible for $p(q_i^{t+1} | x_i; \theta_c)$ is 1, the desired distribution over $q_i \in \{1, 2, \ldots N\}$ at iteration $t+1$ for each example is given by,

$$p_0^{t+1}(q_i | x_i) = \begin{cases} 1, & \text{if } q_i = q_i^{t+1} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

Thus the optimization strategy is to minimize the distance between the distributions $p(q_i | x_i; \theta_c)$ and the one given in eq. (15). In the proposed work, we use KL divergence.

$$\theta_c^{t+1} = \arg\min_{\theta_c} \sum_{i=1}^{M} KL(F_i^{t+1} || G_i), \tag{16}$$

where $F_i^{t+1} = p_0^{t+1}(q_i | x_i)$ and $G_i = p(q_i | x_i; \theta_c)$

After hard EM based pre-training, we combine the outputs of the pre-trained DNNs as given in eq. (1) and jointly train the whole system with new objective function given in eq. (2).

3. Experiments

To conduct experiments, we use TIMIT [12] and Aurora 2 [13] databases for clean speech and noise recordings respectively. TIMIT database is divided into train and test categories containing 4620 and 1680 clean speech utterances respectively with sampling rate of 16kHz. The noise recordings are babble, restaurant, street, airport, car, exhibition, subway and train with sampling rate of 8kHz. Hence, we down sample clean speech recordings to 8kHz. In addition, we use additive white Gaussian (AWGN) noise for the experiments.

We use the noise recordings namely babble, restaurant, street and AWGN (these four noises are seen noises and remaining noises are unseen noises) for training and validation purposes following the work in [18]. All the utterances in train category in the TIMIT database are added with the above mentioned noise types at four different SNR levels, -5 dB, 0 dB, 5 dB and 10 dB. The frame length and frame shift are set to be 256 samples and 128 samples respectively. From the resulting frames, we randomly select 100k examples per configuration (each noise type at particular SNR). Thus we have a total of $10^5 \times 4 \times 4$ frames. We divide this data in $8:2$ ratio (From each configuration, 80k and 20k frames) for training and validation. The testing is done on 250 TIMIT test sentences under above mentioned training noise cases at different SNRs. To test the generalization performance, we also evaluate on the remaining five noise types (not used in training).

We evaluate our method with the number of DNNs in dMoE set to be $N = 2$ with each DNN having three hidden layers. The number of units at each layer is set to be 1024. We use relu activation at the hidden layers and output activation is linear. We refer this system as M-DNN$_{P2}$. In similar way, we build a soft EM based pre-trained dMoE and also dMoE without pre-training of individual DNNs. These two systems are referred as M-DNN$_{S2}$ and M-DNN$_{J2}$ respectively.

To compare methods performance under different number of DNNs we implement proposed method with $N = 4$, & 8 in addition to $N = 2$. These variants are referred as M-DNN$_{P4}$ and M-DNN$_{P8}$ respectively. To maintain total number of parameters similar as compared to $N = 2$ system, we set the number of units at each hidden layer to be 512 and 256 respectively for $N = 4$ and $N = 8$. Note that, the gating network in

	seen cases												unseen cases											
	PESQ				seg SNR (dB)				STOI				PESQ				seg SNR (dB)				STOI			
	-5 dB	0 dB	5 dB	10 dB	-5 dB	0 dB	5 dB	10 dB	-5 dB	0 dB	5 dB	10 dB	-5 dB	0 dB	5 dB	10 dB	-5 dB	0 dB	5 dB	10 dB	-5 dB	0 dB	5 dB	10 dB
S-DNN$_1$	2.19	2.53	2.79	2.99	0.47	1.97	3.41	4.68	0.708	0.796	0.853	0.888	1.78	2.10	2.41	2.72	-2.93	-0.67	1.51	3.36	0.587	0.718	0.816	0.876
S-DNN$_2$	2.19	2.53	2.77	2.96	0.39	1.92	3.40	4.73	0.717	0.799	0.853	0.886	1.76	2.08	2.39	2.69	-3.01	-0.76	1.50	3.51	0.580	0.715	0.815	0.875
M-DNN$_{J2}$	2.22	2.57	2.84	3.05	0.59	2.10	3.59	4.93	0.718	0.804	0.862	0.898	1.78	2.10	2.42	2.74	**-2.79**	-0.66	1.53	3.52	**0.591**	0.720	0.821	0.884
M-DNN$_{S2}$	2.22	2.56	2.83	3.04	0.54	2.05	3.49	4.81	0.716	0.804	0.864	0.900	1.77	2.08	2.40	2.73	-2.84	-0.70	1.44	3.33	0.587	0.720	0.823	0.887
M-DNN$_{P2}$	**2.24**	**2.59**	**2.87**	**3.11**	**0.70**	**2.39**	**4.08**	**5.70**	**0.720**	**0.810**	**0.869**	**0.907**	**1.79**	**2.11**	**2.44**	**2.79**	-2.96	**-0.54**	**1.94**	**4.19**	0.590	**0.725**	**0.829**	**0.894**
M-DNN$_{P4}$	2.23	2.58	2.86	3.10	0.68	2.34	4.00	5.49	**0.721**	0.810	0.869	0.907	1.77	2.11	2.45	2.78	-2.85	-0.48	1.90	4.00	0.592	0.727	0.830	0.894
M-DNN$_{P8}$	2.18	2.55	2.86	3.11	0.57	2.28	3.98	5.58	0.718	0.808	0.869	0.907	1.77	2.11	**2.46**	**2.80**	-2.83	**-0.46**	**1.97**	4.16	**0.594**	**0.728**	**0.831**	0.894

Table 1: *Comparison of proposed method with baselines for seen and unseen noise cases. Blue numbers indicate the best performing scheme. Blue numbers indicate the best performing scheme. We also present evaluation of proposed method with $N = 4, \& 8$. The entries in the last two rows are marked in red if either of them achieves better performance than M-DNN$_{P2}$.*

the above mentioned different dMoEs has similar architecture as that of individual experts except the activation at the output layer is set to be SoftMax.

A single DNN based baseline is implemented following the work of Xu et al. The DNN consists of three hidden layers with 1024 units at each layer. We refer single DNN based enhancement scheme as S-DNN$_1$. We also implement single DNN with number of units at each layer set to be 2048, to make sure that it has similar number of parameters as that of proposed multiple DNN system. This variant is referred as S-DNN$_2$. We did not report SPP based speech enhancement performance as it performed poorly in most of the SNR cases considered.

We use batch normalization [7] and dropout [19] (with $p = 0.2$) between the hidden layers. The number of epochs for each DNN training is set to be 50 with early stopping criteria [20]. Note that the total number of epochs includes both pre-training followed by joint training (no. of epochs for pre-training + no. of epochs for joint training = 50) for hard EM and soft EM based approaches. However, for M-DNN$_{J2}$, S-DNN$_1$ and S-DNN$_2$ systems, it is 50 for the entire joint training (no pre-training). We use adam optimizer with default parameters [21] for the optimization. The loss function used is *mse*. The input data is normalized to have zero mean and unit variance. All the experiments are implemented in python using a deep learning library called keras [20].

We compare the different schemes in terms of PESQ, seg SNR and STOI scores. PESQ is a measure of perceptual quality of speech, while STOI measures the intelligibility. seg SNR provides information about average reconstruction error across frames with respect to the clean speech. Hence, these measures are used to objectively evaluate the enhanced speech.

4. Results and discussion

The results of different methods considered are shown in Table. 1. The numbers presented here are average values over seen noises and unseen noises at different SNRs considered. For seen noise cases, we see that the proposed method (M-DNN$_{P2}$) outperforms the baselines considered for input SNRs -5, 0, 5, and 10 dB in most cases in terms of PESQ, seg SNR and STOI. In specific, the improvement is more at 5 and 10 dB SNRs. This can be attributed to the fact that at high SNRs the underlying structures are more distinguishable resulting in more accurate output of the classifier network. Note that single DNN system performed better than the proposed method for white noise at SNR -5 dB in terms of objective measures PESQ and seg SNR. In a similar fashion, we observe prominent improvement at 5 and 10dB SNRs for unseen noise cases and the performance is similar to single DNN at -5 dB SNR.

4.1. Dependency on the number of DNNs

The number of DNNs used plays an important role in the performance of the system. As we increase the number of DNNs with the number of units at hidden layers fixed, the complexity of the system increases. Hence, it results in overfitting. However, we can overcome this problem in some cases by reducing the number of units. As a reference we present the performance variation of our method with respect to the number of DNNs ($N = 2, 4, \& 8$) as shown in last three rows of the Table. 1. We observe almost similar results for these three variants. As we increase the number of DNNs further, we didn't see much improvement and we also observe decrease in performance for some cases.

4.2. Dependency on the decay parameter

The decay parameter can be seen as the variance of the predicted output. As we increase the decay parameter the error on the training data decreases but it performs poorly on unseen test data. However, the choice of the decay parameter depends on the expected error range of the objective function. We observe that the suitable range of the decay parameter for the considered objective function is $6 - 8$.

4.3. Training complexity

The proposed method, in effect, divides the data among different DNNs at each iteration. amount of time required is N (the number of experts) times more than hard EM approach, since each experts is trained with the entire data (with weightage for each sample) at each iteration. Thus, the time complexities of the pre-training procedures for soft EM and hard EM are of $\mathcal{O}(N)$ and $\mathcal{O}(1)$ respectively, given the same amount of data.

4.4. The distribution of data points

After hard EM based pre-training of the dMoE, we check how data points corresponding to each phoneme are distributed among the individual DNN experts. We notice that all the noisy frames belonging to white and babble noise at SNR -5 dB are assigned to one DNN. Other than that, we observe no particular pattern in the assignment of data points.

5. Conclusion

We proposed a hard expectation maximization based pre-training method for dMoE. Such system outperforms single DNN schemes and dMoE trained jointly without pre-training. The current work can be extended based on different probabilistic models for each expert that better suits speech data. In addition we can also put constraint on the objective function based on speech knowledge (similar to having a prior). These are parts of our future works.

6. References

[1] P. C. Loizou, *Speech Enhancement: Theory and Practice*. USA:CRC press, 2013.

[2] S. Tamura, "An analysis of a noise reduction neural network," *International Conference on Acoustics, Speech, and Signal Processing,*, vol. 3, pp. 2001–2004, May 1989.

[3] F. Xie and D. V. Compernolle, "A family of MLP based nonlinear spectral estimators for noise reduction," *International Conference on Acoustics, Speech, and Signal Processing,*, vol. 2, pp. II/53–II/56, Apr 1994.

[4] E. A. Wan and A. T. Nelson, "Networks for speech enhancement," *Handbook of neural networks for speech processing. Artech House, Boston, USA*, vol. 139, 1999.

[5] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[6] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *Aistats*, vol. 9, 2010, pp. 249–256.

[7] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[8] A. Fischer and C. Igel, "An introduction to restricted boltzmann machines," *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 14–36, 2012.

[9] A. Das and J. H. L. Hansen, "Phoneme selective speech enhancement using parametric estimators and the mixture maximum model: A unifying approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2265–2279, Oct 2012.

[10] S. E. Chazan, S. Gannot, and J. Goldberger, "A phoneme-based pre-training approach for deep neural network with application to speech enhancement," *IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 1–5, Sept 2016.

[11] S. E. Chazan, J. Goldberger, and S. Gannot, "Speech enhancement using a deep mixture of experts," *CoRR*, vol. abs/1703.09302, 2017. [Online]. Available: http://arxiv.org/abs/1703.09302

[12] J. S. Garofolo *et al.*, "Getting started with the DARPA TIMIT CD-ROM: An acoustic phonetic continuous speech database," *National Institute of Standards and Technology (NIST), Gaithersburgh, MD*, vol. 107, 1988.

[13] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," *Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.

[14] K. P. A. Ksentini, C. Viho, and J. Bonnin, "Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs," *PhD/Masters Thesis, University of Rennes*, 2009.

[15] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4214–4217.

[16] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.

[17] D. McAllester, "Lecture notes in statistical methods for artificial intelligence - k means and expectation maximization," http://ttic.uchicago.edu/~dmcallester/ttic101-07/lectures/em/em.pdf, Autumn 2007.

[18] Y. Xu, J. Du, L. R. Dai, and C. H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, Jan 2014.

[19] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012. [Online]. Available: http://arxiv.org/abs/1207.0580

[20] F. Chollet, "Keras," https://github.com/fchollet/keras, 2017.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980