CSLT TECHNICAL REPORT-20150001 [Saturday 10th January, 2015]

Noisy Training for Deep Neural Networks in Speech Recognition

Shi Yin[1,4], Chao Liu[1,3], Zhiyong Zhang[1,2], Yiye Lin[1,5], Dong Wang[1,2*], Javier Tejedor[6], Thomas Fang Zheng[1,2] and Yinguo Li[4]

*Correspondence: wang-dong99@mails.tsinghua.edu.cn
[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China
Full list of author information is available at the end of the article

Abstract

Deep neural networks (DNN) have gained remarkable success in speech recognition, partially attributed to the flexibility of DNN models in learning complex patterns of speech signals. This flexibility, however, may lead to serious over-fitting and hence miserable performance degradation in adverse acoustic conditions such as those with high ambient noises. We propose a noisy training approach to tackle this problem: by injecting moderate noises into the training data intentionally and randomly, more generalizable DNN models can be learned. This 'noise injection' technique, although known to the neural computation community already, has not been studied with DNNs which involve a highly complex objective function. The experiments presented in this paper confirm that the noisy training approach works well for the DNN model and can provide substantial performance improvement for DNN-based speech recognition.

Keywords: speech recognition; deep neural network; noise injection

1 Introduction

A modern automatic speech recognition (ASR) system involves three components: an acoustic feature extractor to derive representative features for speech signals, an emission model to represent static properties of the speech features, and a transitional model to depict dynamic properties of speech production. Conventionally, the dominant acoustic features in ASR are based on short-time spectral analysis, e.g., Mel frequency cepstral coefficients (MFCC). The emission and transition models are often chosen to be the Gaussian mixture model (GMM) and the hidden Markov model (HMM), respectively.

Deep neural networks (DNN) have gained brilliant success in many research fields including speech recognition, computer vision (CV), and natural language processing (NLP) [1]. A DNN is a neural network (NN) that involves more than one hidden layer. NNs have been studied in the ASR community for a decade, mainly in two approaches: in the 'hybrid approach', the NN is used to substitute for the GMM to produce frame likelihood [2], and in the 'tandem approach', the NN is used to produce long-context features that are used to substitute for or augment to short-time features, e.g., MFCCs [3].

Although promising, the NN-based approach, either by the hybrid setting or the tandem setting, did not deliver overwhelming superiority over the conventional approaches based on MFCCs and GMMs. The revolution took place in 2010 after the close collaboration between academic and industrial research groups, including

University of Toronto, Microsoft, and IBM [1, 4, 5]. This research found that very significant performance improvements can be accomplished with the NN-based hybrid approach, with a few novel techniques and design choices: (1) extending NNs to DNNs, i.e., involving a large number of hidden layers (usually 4-8); (2) employing appropriate initialization methods, e.g., pre-training with restricted Boltzmann machines (RBMs); (3) using fine-grained NN targets, e.g., context-dependent states. Since then, numerous experiments have been published to investigate various configurations of the DNN-based acoustic modeling, and all the experiments confirmed that the new model is predominantly superior to the classical architecture based on GMMs [2, 4, 6, 7, 8, 9, 10, 11, 12, 13].

Encouraged by the success of DNNs in the hybrid approach, researchers reevaluated the tandem approach using DNNs and achieved similar performance improvements [3, 14, 15, 16, 17, 18, 19, 20]. Some comparative studies were conducted for the hybrid and tandem approaches, though no evidence supports that one approach clearly outperforms the other [21, 22]. The study of this paper is based on the hybrid approach, though the developed technique can be equally applied to the tandem approach.

The advantage of DNNs in modeling state emission distributions, when compared to the conventional GMM, has been discussed in some previous publications, e.g., [1, 2]. Although no full consentience exists, researchers agree on some points, e.g., the DNN is naturally discriminative when trained with an appropriate objective function, and it is a hierarchical model that can learn patterns of speech signals from primitive levels to high levels. Particularly, DNNs involve very flexible and compact structures: it usually consists of a large amount of parameters and the parameters are highly shared among feature dimensions and task targets (phones or states). This flexibility, on one hand, leads to very strong discriminative models, and on the other hand, may cause serious over-fitting problems, leading to miserable performance reduction if the training and test conditions are mismatched. For example, when the training data are mostly clean and the test data are corrupted by noises, ASR performance usually suffers from a substantial degradation. This over-fitting is particularly serious if the training data are not abundant [23].

A multitude of research has been conducted to improve noise robustness of DNN models. The multi-condition training approach was presented in [24], where DNNs were trained by involving speech data in various channel/noise conditions. This approach is straightforward and usually delivers good performance, though collecting multi-condition data is not always possible. Another direction is to use noise-robust features, e.g., auditory features based on Gammatone filters [23]. The third direction involves various speech enhancement approaches. For example, the vector Taylor series (VTS) was applied to compensate for input features in an adaptive training framework [25]. The authors of [26] investigated several popular speech enhancement approaches and found that the maximum likelihood spectral amplitude estimator (MLSA) is the best spectral restoration method for DNNs trained with clean speech and tested on noisy data. Some other researches involve noise information in DNN inputs and trains a 'noise aware' network. For instance, [27] used the VTS as the noise estimator to generate noise-dependent inputs for DNNs.

Another related technique is the denoising auto-encoder (DAE) [28]. In this approach, some noises are randomly selected and intentionally injected to the original clean speech; the noise-corrupted speech data are then fed to an auto-encoder (AE) network where the targets (outputs) are the original clean speech. By this configuration, the AE will learn the denoising function in a non-linear way. Note that this approach is not particular for ASR, but a general denoising technique. The authors of [29] extended this approach by introducing recurrent NN structures and demonstrated that the deep and recurrent auto-encoder can deliver better performance for ASR in most of the noise conditions they examined.

This paper presents a noisy training approach for DNN-based ASR. The idea is simple: by injecting some noises to the input speech data when conducting DNN training, the noise patterns are expected to be learned, and the generalization capability of the resulting network is expected to be improved. Both may improve robustness of DNN-based ASR systems within noisy conditions. Note that part of the work has been published in [30], though this paper presents a full discussion of the technique and reports extensive experiments.

The paper is organized as follows: Section 2 discusses some related work, and Section 3 presents the proposed noisy training approach. The implementation details are presented in Section 4, and the experimental settings and results are presented in Section 5. The entire paper is concluded in Section 6.

2 Related work

The noisy training approach proposed in this paper was highly motivated by the noise injection theory which has been known for decades in the neural computing community [31, 32, 33, 34]. This paper employs this theory and contributes in two aspects: first, we examine the behavior of noise injection in DNN training, a more challenging task involving a huge amount of parameters; second, we study mixture of multiple noises at various levels of signal-to-noise ratios (SNR), which is beyond the conventional noise injection theory that assumes small and Gaussian-like injected noises.

Another work related to this study is the DAE approach [28, 29]. Both the DAE and the noisy training approaches corrupt NN inputs by randomly sampled noises. Although the objective of the DAE approach is to recover the original clean signals, the focus of the noisy training approach proposed here is to construct a robust classifier.

Finally, this work is also related to the multi-condition training [24], in the sense that both train DNNs with speech signals in multiple conditions. However, the noisy training obtains multi-conditioned speech data by corrupting clean speech signals, while the multi-condition training uses real-world speech data recorded in multiple noise conditions. More importantly, we hope to set up a theoretical foundation and a practical guideline for training DNNs with noises, instead of just regarding it as a blind noise pattern learner.

3 Noisy training

The basic process of noisy training for DNNs is as follows: first of all, sample some noise signals from some real-world recordings, and then mix these noise signals with the original training data. This operation is also referred to as 'noise injection' or 'noise corruption' in this paper. The noise-corrupted speech data are then used to train DNNs as usual. The rationale of this approach is two-fold: firstly, the noise patterns within the introduced noise signals can be learned and thus compensated for in the inference phase, which is straightforward and shares the same idea as the multi-condition training approach; secondly, the perturbation introduced by the injected noise can improve generalization capability of the resulting DNN, which is supported by the noise injection theory. We discuss these two aspects sequentially in this section.

3.1 Noise pattern learning

The impact of injecting noises in training data can be understood as providing some noise-corrupted instances so that they can be learned by the DNN structure and recognized in the inference (test) phase. From this perspective, the DNN and GMM are of no difference, since both can be benefit from matched acoustic conditions of training and testing, by either re-training or adaptation.

However, the DNN is more powerful in noise pattern learning than the GMM. Due to its discriminative nature, the DNN model focuses on phone/state boundaries, and the boundaries it learns might be highly complex. Therefore, it is capable of addressing more severe noises and dealing with heterogeneous noise patterns. For example, a DNN may obtain a reasonable phone classification accuracy in a very noisy condition, if the noise does not drastically change the decision boundaries (e.g., with car noise). In addition, noises of different types and at different magnitude levels can be learned simultaneously, as the complex decision boundaries that the DNN classifier may learn provides sufficient freedom to address complicated decisions in heterogeneous acoustic conditions.

In contrast, the GMM is a generative model and focuses on class distributions. The decision boundaries a GMM learns (which are determined by the relative locations of the GMM components of phones/states) are relative much simpler than those a DNN model learns. Therefore, it is difficult for GMMs to address heterogeneous noises.

The above argument explains some interesting observations in the DNN-based noise training in our experiments. First, learning a particular type of noise does not necessarily lead to performance degradation in another type of noise. In fact, our experiments show that learning a particular noise usually improves performances on other noises, only if the property of the ‘unknown’ noise is not drastically different from the one that has been learned. This is a clear advantage over GMMs, for which a significant performance reduction is often observed when the noise conditions of training and test data are unmatched.

Moreover, our experiments show that learning multiple types of noises are not only possible, but also complementary. As we will see shortly, learning two noises may lead to better performance than learning any single noise, when the test data are corrupted by either of the two noises. This is also different from GMMs, for which learning multiple noises generally leads to interference among each other.

The power of DNNs in learning noise patterns can be understood in a deeper way, from three perspectives. Firstly, the DNN training is related to feature selection. Due to the discriminative nature, the DNN training can infer the most discriminative part of the noise-corrupted acoustic features. For instance, with the training data corrupted by car noise, the DNN training process will learn that the corruption is mainly on the low frequency part of the signal, and so the low frequency components of the speech features are deemphasized in the car noise condition. Learning the car noise, however, did not seriously impact the decision boundaries in other conditions in our experiments, e.g., with clean speech, probably due to the complicated DNN structure that allows to learn noise-conditioned decision boundaries. Moreover, learning car noise may benefit other noise conditions where the corruption mainly resides in low frequency components (as the car noise), even though the noise is not involved in the training.

Secondly, the DNN training is related to perceptual classification. Thanks to the multi-layer structure, DNNs learn noise patterns gradually. This means that the noise patterns presented to the DNN inputs are learned together with the speech patterns at low levels, but only at high levels, the noise patterns are recognized and deemphasized in the decision. This provides a large space for DNNs to learn heterogeneous noise patterns and 'memorize' them in the abundant parameters. This process also simulates the processing procedure of the human brain, where noise patterns are processed and recognized by the peripheral auditory system but are ignored in the final perceptual decision by the central neural system.

Finally, the DNN training is related to the theory of regularization. All admit that a large amount of parameters of DNNs allow great potential to learn complex speech and noise patterns and their class boundaries. If the training is based on clean speech only, however, the flexibility provided by the DNN structure is largely wasted. This is because the phone class boundaries are relatively clear with clean speech, and so the abundant parameters of DNNs tend to learn the nuanced variations of phone implementations, conditioned on a particular type of channel and/or background noise. This is a notorious over-fitting problem. By injecting random noises, the DNN training is enforced to emphasize on the most discriminative patterns of speech signals. In other words, the DNNs trained with noise injection tend to be less sensitive to noise corruptions. This intuition is supported by the noise injection theory as presented in the next section.

3.2 Noise injection theory

It has been known for two decades that imposing noises to the input can improve generalization capability of neural networks [35]. A bunch of theoretical studies have been presented to understand the implication of this 'noise injection'. Nowadays, it is clear that involving a small magnitude of noise in the input is equivalent to introducing a certain regularization in the objective function, which in turn encourages the network converging to a smoother mapping function [36]. More precisely, with noise injection, the training favors an optimal solution at which the objective function is less sensitive to the change of the input [32]. Further studies showed that noise injection is closely correlated to some other well-known techniques, including sigmoid gain scaling and target smoothing by convolution [37], at least with Gaussian noises and multi-layer perceptrons (MLP) with a single layer. The relationships among regularization, weight decay and noise injection, on one hand, provide a better understanding for each individual technique, and on the other hand, motivate some novel and efficient robust training algorithms. For example, Bishop showed that noise injection can be approximated by a Tikhonov regularization on the square error cost function [33]. Finally, we note that noise injection can be conducted in different ways, such as perturbation on weights and hidden units [31], though we just consider the noise injection to the input in this paper.

In order to highlight the rationale of noise injection (and so noisy training), we reproduce the formulation and derivation in [32], but migrate the derivation to the case of cross entropy cost which is usually used in classification problems such as ASR.

First of all, formulate an MLP as a nonlinear mapping function $f_\theta : \mathcal{R}^M \longmapsto \mathcal{R}^K$ where M is the input dimension and K is the output dimension, and θ encodes all the parameters of the network including weights and biases. Let $\mathbf{x} \in \mathcal{R}^M$ denote the input variables, and $\mathbf{y} \in \{0,1\}^K$ denote the target labels which follow the 1-of-K encoding scheme. The cross entropy cost is defined as follows:

$$E(\theta) = -\sum_{n=1}^{N}\sum_{k=1}^{K}\{\mathbf{y}^{(n)} ln f_k(\mathbf{x}^{(n)})\} \tag{1}$$

where n indexes the training samples and k indexes the output units. Considering an identical and independent noise $\mathbf{v}$ whose first and second moments satisfy the following constraints:

$$\mathbb{E}\{\mathbf{v}\} = 0 \qquad \mathbb{E}\{\mathbf{v}^2\} = \epsilon I \tag{2}$$

where I is the M-dimensional identity matrix, and ϵ is a small positive number. Applying the Taylor series of $lnf(\mathbf{x})$, the cost function with the noise injection can be derived as follows:

$$\begin{aligned} E_v(\theta) &= -\sum_{n=1}^{N}\sum_{k=1}^{K}\{\mathbf{y}_k^{(n)} ln f_k(\mathbf{x}^{(n)} + \mathbf{v}^{(n)})\} \\ &\approx -\sum_{n=1}^{N}\sum_{k=1}^{K}\{\mathbf{y}_k^{(n)} ln f_k(\mathbf{x}^{(n)})\} \\ &- \sum_{n=1}^{N}\sum_{k=1}^{K}\mathbf{y}_k^{(n)}\{\mathbf{v}^{(n)T}\frac{\bigtriangledown f_k(\mathbf{x}^{(n)})}{f_k(\mathbf{x}^{(n)})} + \frac{1}{2}\mathbf{v}^{(n)T}H_k(\mathbf{x}^{(n)})\mathbf{v}^{(n)}\} \end{aligned}$$

where $H_k(x)$ is defined as follows:

$$H_k(x) = \frac{-1}{f_k(\mathbf{x})} \bigtriangledown f_k(\mathbf{x}) \bigtriangledown f_k(\mathbf{x})^T + \frac{1}{f_k^2(\mathbf{x})} \bigtriangledown^2 f_k(\mathbf{x}).$$

Since $\mathbf{v}^{(n)}$ is independent of $\mathbf{x}^{(n)}$ and $\mathbb{E}\{\mathbf{v}\} = 0$, the first order item vanishes and the cost is written as:

$$E_v(\theta) \approx E(\theta) - \frac{\epsilon}{2}\sum_{k=1}^{K} tr(\tilde{H}_k) \tag{3}$$

where tr denotes the trace operation, and

$$\tilde{H}_k = \sum_{n \in \mathcal{C}_k} H_k(\mathbf{x}^{(n)})$$

where $\mathcal{C}_k$ is the set of indices of the training samples belonging to the k-th class.

In order to understand the implication of Eq. (3), an auxiliary function can be defined as follows:

$$E(\theta, \mathbf{v}) = -\sum_{n=1}^{N}\sum_{k=1}^{K}\{\mathbf{y}_k^{(n)} ln f_k(\mathbf{x}^{(n)} + \mathbf{v})\}$$

where $\mathbf{v}$ is a small change to the input vectors $\{\mathbf{x}^{(n)}\}$. Note that $E(\theta, \mathbf{v})$ differs from $E_v(\theta)$: $\mathbf{v}$ in $E(\theta, \mathbf{v})$ is a fixed value for all $\mathbf{x}^{(n)}$, while $\mathbf{v}^{(n)}$ in $E_v(\theta)$ is a random variable and differs for each training sample. The Laplacian of $E(\theta, \mathbf{v})$ with respect to $\mathbf{v}$ is computed as follows:

$$\begin{aligned}
\bigtriangledown^2 E(\theta, \mathbf{v}) &= tr\{\frac{\partial^2 E(\theta, \mathbf{v})}{\partial \mathbf{v}^2}\} \\
&= -tr\{\sum_{n=1}^{N}\sum_{k=1}^{K} \mathbf{y}_k^{(n)} H_k(\mathbf{x}^{(n)} + \mathbf{v})\} \\
&= -tr\{\sum_{k=1}^{K}\sum_{n \in \mathcal{C}_k} H_k(\mathbf{x}^{(n)} + \mathbf{v})\}.
\end{aligned} \tag{4}$$

By comparing Eq. (4) and Eq. (3), we get:

$$E_v(\theta) \approx E(\theta) + \frac{\epsilon}{2} \bigtriangledown^2 E(\theta, 0). \tag{5}$$

Eq. (5) indicates that injecting noises to the input units is equivalent to placing a regularization on the cost function. This regularization is related to the second order derivatives of the cost function with respect to the input, and its strength is controlled by the magnitude of the injected noise. Since $\bigtriangledown^2 E(\theta, 0)$ is positive at the optimal solution of θ, the regularized cost function tends to accept solutions with a smaller curvature of the cost. In other words, the new cost function $E_v(\theta)$ is less sensitive to the change on inputs, and therefore leads to better generalization capability. Note that this result is identical to the one obtained in [32], where the cost function is the square error.

4 Noisy deep learning

From the previous section, the validity of the noisy training approach can be justified in two ways: discriminative noise pattern learning and objective function smoothing. The former provides the ability to learn multiple noise patterns, and the latter encourages a more robust classifier. However, it is still unclear if the noisy training scheme works for the DNN model which involves a large number of parameters and thus tends to exhibit a highly complex cost function. Particularly, the derivation of Eq. (5) assumes small noises with diagonal covariances, while in practice we wish to learn complex noise patterns that may be large in magnitude and fully dimensional correlated. Furthermore, the DNN training is easy to fall in a local minimum, and it is not obvious if the random noise injection may lead to fast convergence.

We therefore investigate how the noise training works for DNNs when the injected noises are large in magnitude and heterogeneous in types. In order to simulate noises in practical scenarios, the procedure illustrated in Fig. 1 is proposed.

For each speech signal (utterance), we first select a type of noise to corrupt it. Assuming that there are n types of noises, we randomly select a noise type following a multinomial distribution:

$$v \sim Mult(\mu_1, \mu_2, ..., \mu_n).$$

The parameters $\{\mu_i\}$ are sampled from a Dirichlet distribution:

$$(\mu_1, \mu_2, ..., \mu_n) \sim Dir(\alpha_1, \alpha_2, ..., \alpha_n)$$

where the parameters $\{\alpha_i\}$ are manually set to control the base distribution of the noise types. This hierarchical sampling approach (Dirichlet followed by multinomial) simulates the uncertain noise type distributions in different operation scenarios. Note that we allow a special noise type 'no-noise', which means that the speech signal is not corrupted.

Secondly, sample the noise level (i.e., SNR). This sampling follows a Gaussian distribution $\mathcal{N}(\mu_{SNR}, \sigma_{SNR})$ where μ_{SNR} and σ_{SNR} are the mean and variance respectively, and are both manually defined. If the noise type is no-noise, then the SNR sampling is not needed.

The next step is to sample an appropriate noise segment according to the noise type. This is achieved following a uniformed distribution, i.e., randomly select a starting point b in the noise recording of the required noise type, and then excerpt a segment of signal which is of the same length as the speech signal to corrupt. Circular excerption is employed if the length of the noise signal is less than the speech signal.

Finally, the selected noise segment is scaled to reach the required SNR level, and then is used to corrupt the clean speech signal. The noise-corrupted speech is fed into the DNN input units to conduct model training.

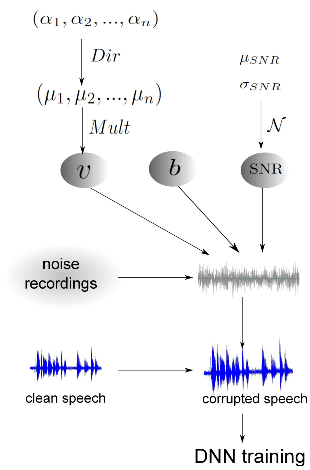


Figure 1 The noise training procedure. 'Dir' denotes the Dirichlet distribution, '$Mult$' denotes the multinomial distribution, and '$\mathcal{N}$' denotes the Gaussian distribution. v is a variable that represents the noise type, b represents the starting frame of the selected noise segment, and SNR is the expected SNR of the corrupted speech data.

5 Experiments

5.1 Databases

The experiments were conducted with the Wall Street Journal (WSJ) database. The setting is largely standard: the training part used the WSJ si284 training dataset, which involves 37318 utterances or about 80 hours of speech signals. The WSJ dev93 dataset (503 utterances) was used as the development set for parameter tuning and cross validation in DNN training. The WSJ eval92 dataset (333 utterances) was used to conduct evaluation.

Note that the WSJ database was recorded in a noise-free condition. In order to simulate noise-corrupted speech signals, the DEMAND noise database[1] was used to sample noise segments. This database involves 18 types of noises, from which we selected 7 types in this work, including white noise and noises at cafeteria, car, restaurant, train station, bus and park.

5.2 Experimental settings

We used the Kaldi toolkit[2] to conduct the training and evaluation, and largely followed the WSJ s5 recipe for Graphics Processing Unit (GPU)-based DNN training. Specifically, the training started from a monophone system with the standard 13-dimensional MFCCs plus the first and second order derivatives. Cepstral mean normalization (CMN) was employed to reduce the channel effect. A triphone system was then constructed based on the alignments derived from the monophone system, and a Linear Discriminant Analysis (LDA) transform was employed to select the most discriminative dimensions from a large context (5 frames to the left and right respectively). A further refined system was then constructed by applying a maximum likelihood linear transform (MLLT) upon the LDA feature, which intended to reduce the correlation among feature dimensions so that the diagonal assumption of

[1] http://parole.loria.fr/DEMAND/
[2] http://kaldi.sourceforge.net/

the Gaussians is satisfied. This MLLT+LDA system involves 351 phones and 3447 Gaussian mixtures, and was used to generate state alignments.

The DNN system was then trained utilizing the alignments provided by the MLLT+LDA GMM system. The feature used was 40-dimensional filter banks. A symmetric 11-frame window was applied to concatenate neighboring frames, and an LDA transform was used to reduce the feature dimension to 200. The LDA-transformed features were used as the DNN input.

The DNN architecture involves 4 hidden layers and each layer consists of 1200 units. The output layer is composed of 3447 units, equal to the total number of Gaussian mixtures in the GMM system. The cross entropy was set as the objective function of the DNN training, and the stochastic gradient descendent (SGD) approach was employed to perform optimization, with the mini batch size set to 256 frames. The learning rate started from a relatively large value (0.008), and was then gradually shrunk by halving the value whenever no improvement on frame accuracy on the development set was obtained. The training stopped when the frame accuracy improvement on the cross validation data was marginal (less than 0.001). Neither momentum nor regularization was used, and no pre-training was employed since we did not observe clear advantage by involving these techniques.

In order to inject noises, the averaged energy was computed for each training/test utterance, and a noise segment was randomly selected and scaled according to the expected SNR; the speech and noise signals were then mixed by simple time-domain addition. Note that the noise injection was conducted before the utterance-based CMN. In the noisy training, the training data were corrupted by the selected noises, while the development data used for cross validation remained uncorrupted. The DNNs reported in this section were all initialized from scratch and were trained based on the same alignments provided by the LDA+MLLT GMM system. Note that the process of the model training is reproducible in spite of the randomness on noise injection and model initialization, since the random seed was hard-coded.

In the test phase, the noise type and SNR are all fixed so that we can evaluate the system performance in a specific noise condition. This is different from the training phase where both the noise type and SNR level can be random. We choose the 'big dict' test case suggested in the Kaldi WSJ recipe, which is based on a large dictionary consisting of 150k English words and a corresponding 3-gram language model.

Table 1 presents the baseline results, where the DNN models were trained with clean speech data, and the test data were corrupted with different types of noises at different SNRs. The results are reported in word error rates (WER) on the evaluation data. We observe that without noise, a rather high accuracy (4.31%) can be obtained; with noise interference, the performance is dramatically degraded, and more noise (a smaller SNR) results in more serious degradation. In addition, different types of noises impact the performance in different degrees: the white noise is the most serious corruption which causes a 10 times of WER increase when the SNR is 10dB; in contrast, the car noise is the least impactive: It causes a relatively small WER increase (37% in relative) even if the SNR goes below 5dB.

The different behaviors in WER changes can be attributed to the different patterns of corruptions with different noises: white noise is broad-band and so it corrupts

Table 1 WER of the baseline system.

	WER%					
Test SNR(dB)	5	10	15	20	25	clean
White	77.23	46.46	21.21	9.30	5.51	4.31
Car	5.94	5.42	4.87	4.77	4.50	4.31
Cafeteria	25.33	14.27	10.07	8.38	6.88	4.31
Restaurant	46.87	22.15	13.27	9.73	7.48	4.31
Train Station	34.36	12.72	6.93	5.40	4.43	4.31
Bus	13.88	8.44	6.57	5.51	4.84	4.31
Park	22.10	11.25	7.44	5.87	4.63	4.31

speech signals on all frequency components; in contrast, most of the color noises concentrate on a limited frequency band and so lead to limited corruptions. For example, car noise concentrates on low frequencies only, leaving most of the speech patterns uncorrupted.

5.3 Single noise injection

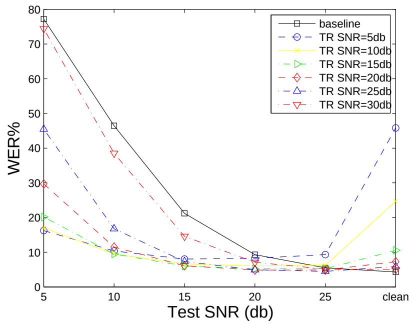
In the first set of experiments, we study the simplest configuration for the noisy training, which is a single noise injection at a particular SNR. This is simply attained by fixing the injected noise type and selecting a small σ_{SNR} so that the sampled SNRs concentrate on the particular level μ_{SNR}. In this section, we choose σ_{SNR}=0.01.

5.3.1 White noise injection

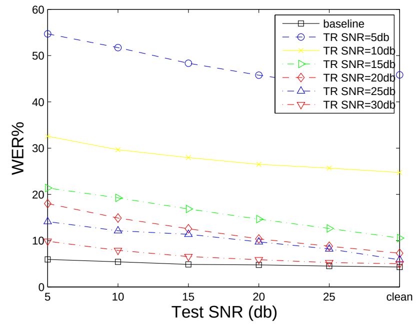
We first investigate the effect of white noise injection. Among all the noises, the white noise is rather special: it is a common noise that we encounter every day, and it is broad-band and often leads to drastic performance degradation compared to other narrow-band noises, as has been shown in the previous section. Additionally, the noise injection theory discussed in Section 3 shows that white noise satisfies Eq. (2) and hence leads to the regularized cost function of Eq. (5). This means that injecting white noise would improve generalization capability of the resulting DNN model; this is not necessarily the case for most of other noises.

Fig. 2 presents the WER results, where the white noise is injected during training at SNR levels varying from 5dB to 30dB, and each curve represents a particular SNR case. The first plot shows the WER results on the evaluation data that are corrupted by white noise at different SNR levels from 5d to 25dB. For comparison, the results on the original clean evaluation data are also presented. It can be observed that injecting white noise generally improves ASR performance on noisy speech, and a matched noise injection (at the same SNR) leads to the most significant improvement. For example, injecting noise at an SNR of 5dB is the most effective for the test speech at an SNR of 5dB, while injecting noise at an SNR of 25dB leads to the best performance improvement for the test speech at an SNR of 25dB. A serious problem, however, is that the noise injection always leads to performance degradation on clean speech. For example, the injection at an SNR of 5dB, although very effective for highly noisy speech (SNR $<$ 10dB), leads to a WER 10 times higher than the original result on the clean evaluation data.

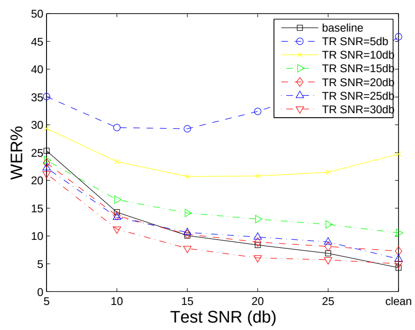
The second and third plots show the WER results on the evaluation data that are corrupted by car noise and cafeteria noise respectively. In other words, the injected noise in training does not match the noise condition in test. It can be seen that



(a) White noise test



(b) Car noise test



(c) Cafeteria noise test

Figure 2 Performance of noisy training with white noise injected ($\sigma = 0.01$). 'TR' means the training condition. The 'baseline' curves present the results of the system trained with clean speech data, as have been presented in Table 1.

the white noise injection leads to some performance gains on the evaluation speech corrupted by the cafeteria noise, as far as the injected noise is limited in magnitude. This demonstrated that the white noise injection can improve the generalization capability of the DNN model, as predicted by the noise injection theory in Section 3. For the car noise corruption, however, the white noise injection does not show any benefit. This is perhaps attributed to the fact that the cost function Eq. (1) is not so bumpy with respect to the car noise, and hence the regularization term introduced in Eq. (3) is less effective. This conjecture is supported by the baseline results which show very little performance degradation with the car noise corruption.

In both the car and cafeteria noise conditions, if the injected white noise is too strong, then the ASR performance is drastically degraded. This is because a strong white noise injection does not satisfy the small noise assumption of Eq. (2) and hence the regularized cost Eq. (3) does not hold anymore. This, on one hand, breaks the theory of noise injection so that the improved generalization capability is not guaranteed, and on the other hand, it results in biased learning towards the white noise-corrupted speech patterns that are largely different from the ones that are observed in speech signals corrupted by noises of cars and cafeterias.

As a summary, white noise injection is effective in two ways: for white noise-corrupted test data, it can learn white noise-corrupted speech patterns and provides dramatic performance improvement particularly at matched SNRs; for test data corrupted by other noises, it can deliver a more robust model if the injection is in a small magnitude, especially for noises that cause a significant change on the DNN cost function. An aggressive white noise injection (with a large magnitude) usually leads to performance reduction on test data corrupted by color noises.

5.3.2 Color noise injection

Besides white noise, in general any noise can be used to conduct the noisy training. We choose the car noise and the cafeteria noise in this experiment to investigate the color noise injection. The results are shown in Fig. 3 and Fig. 4 respectively.

For the car noise injection (Fig. 3), we observe that it is not effective for the white noise-corrupted speech. However, for the test data corrupted by car noise and cafeteria noise, it indeed delivers performance gains. The results with the car noise-corrupted data show clear advantage with matched SNRs, i.e., with the training and test data corrupted by the same noise at the same SNR, the noise injection tends to deliver better performance gains. For the cafeteria noise-corrupted data, it shows that a mild noise injection (SNR=10dB) performs the best. This indicates that there are some similarities between car noise and cafeteria noise, and learning patterns of car noise is useful to improve robustness of the DNN model against corruptions caused by cafeteria noise.

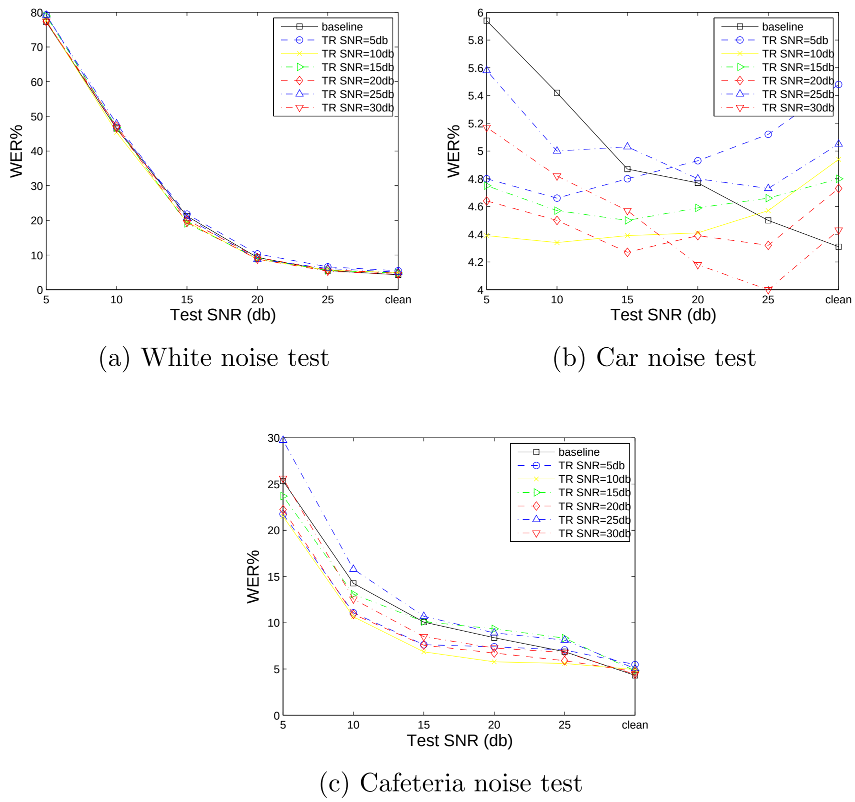
For the cafeteria noise injection (Fig. 4), some improvement can be attained with data corrupted by both white noise and cafeteria noise. For the car noise-corrupted data, performance gains are found only with mild noise injections. This suggests that cafeteria noise possesses some similarities to both white noise and car noise: It involves some background noise which is generally white, and some low frequency components that resemble car noise. Without surprise, the most performance improvement is attained with data corrupted by cafeteria noise.

5.4 Multiple noise injection

In the second set of experiments, multiple noises are injected when performing noisy training. For simplicity, we fix the noise level at SNR=15dB, which is obtained by setting $\mu_{SNR} = 15$ and $\sigma_{SNR} = 0.01$. The hyperparameters $\{\alpha_i\}$ in the noise-type sampling are all set to 10, which generates a distribution on noise types roughly concentrated in the uniform distribution but with a sufficiently large variation.

The first configuration injects white noise and car noise, and test data are corrupted by all the 7 noises. The results in terms of absolute WER reduction are presented in plot (a) of Fig. 5. It can be seen that with the noisy training, almost all the WER reductions (except in the clean speech case) are positive, which means that the multiple noise injection improves the system performance in almost all the noise conditions. An interesting observation is that this approach delivers general good performance gains for the unknown noises, i.e., the noises other than the white noise and the car noise.

The second configuration injects white noise and cafeteria noise; again the conditions with all the 7 noises are tested. The results are presented in plot (b) of Fig. 5. We observe a similar pattern as in the case of white+car noise (plot (a)): The performance on speech corrupted by any noise is significantly improved. The



(a) White noise test

(b) Car noise test

(c) Cafeteria noise test

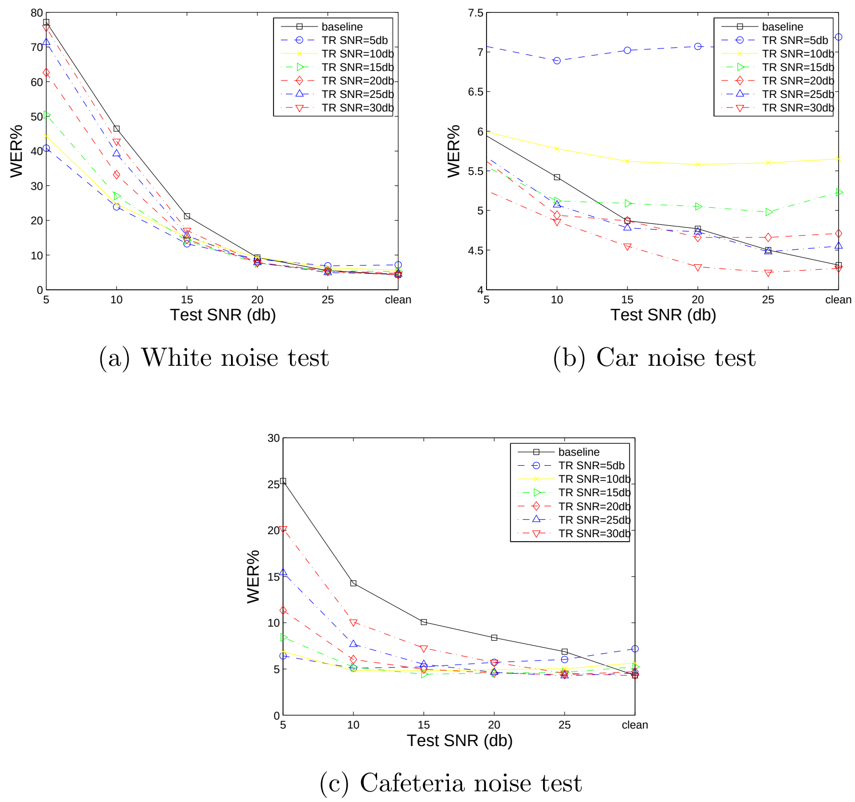
Figure 3 Performance of noisy training with car noise injected ($\sigma = 0.01$). 'TR' means the training condition. The 'baseline' curves present the results of the system trained with clean speech data, as have been presented in Table 1.

difference from plot (a) is that the performance on the speech corrupted by cafeteria noise is more effectively improved, while the performance on the speech corrupted by car noise is generally decreased. This is not surprising as the cafeteria noise is now 'known' and the car noise becomes 'unknown'. Interestingly, the performance on speech corrupted by the restaurant noise and the station noise are both improved in a more effective way than in plot (a). This suggests that the cafeteria noise shares some patterns with these two types of noises.

As a summary, the noisy training based on multiple noise injection is effective in learning patterns of multiple noise types, and it usually leads to significant improvement of ASR performance on speech data corrupted by the noises that have been learned. This improvement, interestingly, can be well generalized to unknown noises. In all the 7 investigated noises, the behavior of the car noise is abnormal, which suggests that car noise is unique in properties and is better to be involved in noisy training.

5.5 Multiple noise injection with clean speech

An obvious problem of the previous experiments is that the performance on clean speech is generally degraded with noisy training. A simple approach to alleviate the



(a) White noise test

(b) Car noise test

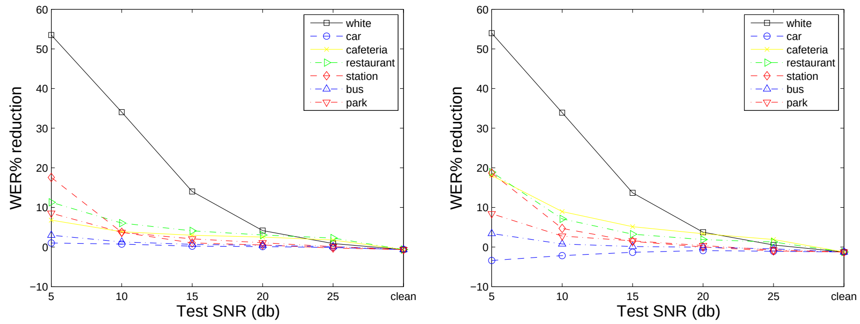
(c) Cafeteria noise test

Figure 4 Performance of noisy training with cafeteria noise injected ($\sigma = 0.01$). 'TR' means the training condition. The 'baseline' curves present the results of the system trained with clean speech data, as have been presented in Table 1.

problem is to involve clean speech in the training. This can be achieved by sampling a special 'no-noise' type together with other noise types. The results are reported in Fig. 6, where plot (a) presents the configuration with white+car noise and plot (b) presents the configuration with white+cafeteria noise. We can see that with clean speech involved in the noisy training, the performance degradation on clean speech is largely solved.

Interestingly, involving clean speech in the noisy training improves performance not only on clean data, but also on noise-corrupted data. For example, plot (b) shows that involving clean speech leads to general performance improvement on test data corrupted by car noise, which is quite different from the results shown in plot (b) of Fig. 5, where clean speech is not involved in the training and the performance on speech corrupted by car noise is actually decreased. This interesting improvement on noise data is maybe due to the 'no-noise' data that provide information about the 'canonical' patterns of speech signals, with which the noisy training is easier to discover the invariant and discriminative patterns that are important for recognition on both clean and corrupted data.

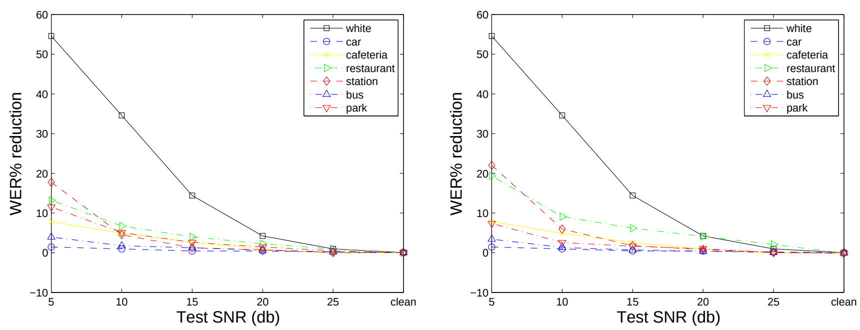
We note that the noisy training with multiple noise injection resembles the multi-condition training: Both involve training speech data under multiple noise condi-



(a) White & car noise

(b) White & cafeteria noise

Figure 5 Performance of multiple noise injection. No clean speech is involved in training.



(a) White & car noise

(b) White & cafeteria noise

Figure 6 Performance of multiple noise injection with clean speech involved in training.

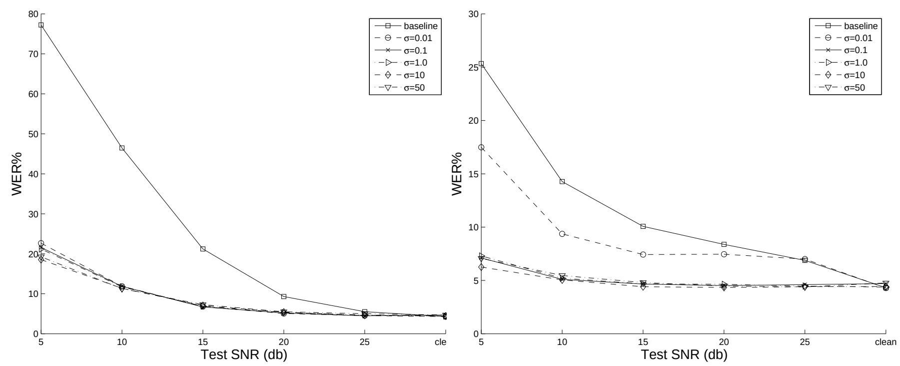
tions. However, there is an evident difference between the two approaches: In multi-conditional training, the training data are recorded under multiple noise conditions and the noise is unchanged across utterances of the same session; in noisy training, noisy data are synthesized by noise injection, so it is more flexible in noise selection and manipulation, and the training speech data can be utilized more efficiently.

5.6 Noise injection with diverse SNRs

The flexibility of noisy training in noise selection can be further extended by involving multiple SNR levels. By involving noise signals at various SNRs, more abundant noise patterns can be learned. More importantly, we hypothesize that the abundant noise patterns provide more negative learning examples for DNN training, so the 'true speech patterns' can be better learned.

The experimental setup is the same as the previous experiment, i.e., fixing μ_{SNR}=15dB and then injecting multiple noises including 'non-noise' data. In order to introduce diverse SNRs, σ_{SNR} is set to be a large value. In this study, σ_{SNR} varies from 0.01 to 50. A larger σ_{SNR} leads to more diverse noise levels and higher possibility for loud noises. For simplicity, only the results with white+cafeteria noise injection are reported, while other configurations were experimented and the conclusions are similar.

Firstly, we examine the performance with ‘known noises’, i.e., data corrupted by white noise and cafeteria noise. The WER results are shown in Fig. 7, where plot (a) presents the results on the data corrupted by white noise, and plot (b) presents the results on the data corrupted by cafeteria noise. We can observe that with a more diverse noise injection (a larger σ_{SNR}), the performances under both the two noise conditions are generally improved. However, if σ_{SNR} is over large, the performance might be decreased. This can be attributed to the fact that a very large σ_{SNR} results in a significant proportion of extremely large or small SNRs, which is not consistent with the test condition. The experimental results show that the best performance is obtained with $\sigma_{SNR} = 10$.



(a) White noise (b) Cafeteria noise

Figure 7 Performance of noise training with different σ_{SNR}. White and Cafeteria noises are injected, and μ_{SNR}=15dB. For each plot, the test data are corrupted by a particular ‘known’ noise. The ‘baseline’ curves present the results of the system trained with clean speech data, as have been presented in Table 1.

In another group of experiments, we examine performance of the noisy-trained DNN model on data corrupted by ‘unknown noises’, i.e., noises that are different from the ones injected in training. The results are reported in Fig. 8. We observe quite different patterns for different noise corruptions: For most noise conditions, we observe a similar trend as in the known noise condition. When injecting noises at more diverse SNRs, the WER tends to be decreased, but if the noise is over diverse, the performance may be degraded. The maximum σ_{SNR} should not exceed 0.1 in most cases (restaurant noise, park noise, station noise). For the car noise condition, the optimal σ_{SNR} is 0.01, and for the bus noise condition, the optimal σ_{SNR} is 1.0. The smaller optimal σ_{SNR} in the car noise condition indicates again that this noise is significantly different from the injected white and cafeteria noises; on the contrary, the larger optimal σ_{SNR} in the bus noise condition suggests that the bus noise resembles the injected noises.

In general, the optimal values of σ_{SNR} in the condition of unknown noises are much smaller than those in the condition of known noises. This is somewhat expected, since injection of over diverse/loud noises that are different from those observed in test tends to cause acoustic mismatch between the training and test data, which may offset the improved generalization capability offered by the noisy training. Therefore, to accomplish the most possible gains with the noisy training, the best strategy is to involve noise types as many as possible in training so that (1) most of the noises in test are known or partially known, i.e., similar noises involved in training; (2) a larger σ_{SNR} can be safely employed to obtain better performance. For a system that operates in unknown noise conditions, the most reasonable strategy is to involve some typical noise types (e.g., white noise, car noise, cafeteria noise) and choose a moderate noise corruption level, i.e., a middle-level μ_{SNR} not larger than 15dB and a small σ_{SNR} not larger than 0.1.

6 Conclusions

We proposed a noisy training approach for DNN-based speech recognition. The analysis and experiments confirmed that by injecting a moderate level of noise in the training data, the noise patterns can be effectively learned and the generalization capability of the learned DNNs can be improved. Both the two advantages result in substantial performance improvement for DNN-based ASR systems in noise conditions. Particularly, we observe that the noisy training approach can effectively learn multiple types of noises, and the performance is generally improved by involving a proportion of clean speech. Finally, noise injection at a moderate range of SNRs delivers further performance gains. The future work involves investigating various noise injection approaches (e.g., weighted noise injection) and evaluating more noise types.

Acknowledgement

This research was supported by the National Science Foundation of China (NSFC) under the project No. 61371136, and the MESTDC PhD Foundation Project No. 20130002120011. It was also supported by Sinovoice and Huilan Ltd.

Author details

[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [2]Center for Speech and Language Technologies, Division of Technical Innovation and Development, Tsinghua National Laboratory for Information Science and Technology, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [3]Department of Computer Science and Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [4]School of Compute Science and Technology, Chongqing University of Posts and Telecommunications (CUPT), No.2, Chongwen Road, Nan'an district, 400065 Chong Qing, China. [5]Beijing Institute of Technology, No.5, South street, Zhonggunacun, Haidian district, 100081 Beijing, China. [6]GEINTRA, University of Alcalá, Spain.

References

1. Li Deng and Dong Yu, *DEEP LEARNING: Methods and Applications*, NOW Publishers, January 2014.
2. Hervé Bourlard and Nelson Morgan, "Hybrid HMM/ANN systems for speech recognition: Overview and new research directions," in *Adaptive Processing of Sequences and Data Structures*, pp. 389–417. Springer, 1998.
3. Hynek Hermansky, Daniel PW Ellis, and Sangita Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2000, pp. 1635–1638.
4. George E Dahl, Dong Yu, Li Deng, and Alex Acero, "Large vocabulary continuous speech recognition with context-dependent DBN-HMMs," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4688–4691.
5. Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
6. A. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *Proc. of Neural Information Processing Systems (NIPS) Workshop Deep Learning for Speech Recognition and Related Applications*, 2009.
7. George E Dahl, Dong Yu, Li Deng, and Alex Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.
8. Dong Yu, Li Deng, and G Dahl, "Roles of pre-training and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition," in *Proc. of NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.
9. Navdeep Jaitly, Patrick Nguyen, Andrew W Senior, and Vincent Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *Proc. of Interspeech*, 2012, pp. 2578–2581.
10. Tara N Sainath, Brian Kingsbury, Bhuvana Ramabhadran, Petr Fousek, Petr Novak, and Abdel-rahman Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011, pp. 30–35.
11. Tara N Sainath, Brian Kingsbury, Hagen Soltau, and Bhuvana Ramabhadran, "Optimization techniques to improve training speed of deep belief networks for large speech tasks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 2267–2276, 2013.
12. Frank Seide, Gang Li, and Dong Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. of Interspeech*, 2011, pp. 437–440.
13. Frank Seide, Gang Li, Xie Chen, and Dong Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011, pp. 24–29.
14. Oriol Vinyals and Suman V Ravuri, "Comparing multilayer perceptron to deep belief network tandem features for robust ASR," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4596–4599.
15. Dong Yu and Michael L Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Proc. of Interspeech*, 2011, pp. 237–240.
16. Peter Bell, Pawel Swietojanski, and Steve Renals, "Multi-level adaptive networks in tandem and hybrid ASR systems," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6975–6979.
17. Frantisek Grezl and Petr Fousek, "Optimizing bottle-neck features for LVCSR," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 4729–4732.
18. Partha Lal and Simon King, "Cross-lingual automatic speech recognition using tandem features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2506–2515, 2011.
19. Christian Plahl, Ralf Schlüter, and Hermann Ney, "Hierarchical bottle neck features for LVCSR," in *Proc. of Interspeech*, 2010, pp. 1197–1200.
20. Tara N Sainath, Brian Kingsbury, and Bhuvana Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4153–4156.
21. Zoltán Tüske, Ralf Schlüter, Hermann Ney, and Martin Sundermeyer, "Context-dependent MLPs for LVCSR: Tandem, hybrid or both?," in *Proc. of Interspeech*, 2012, pp. 18–21.

22. David Imseng, Petr Motlicek, Philip N Garner, and Hervé Bourlard, "Impact of deep MLP architecture on different acoustic modeling techniques for under-resourced speech recognition," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013, pp. 332–337.
23. Jun Qi, Dong Wang, Ji Xu, and Javier Tejedor, "Bottleneck features based on gammatone frequency cepstral coefficients," in *Proc. of Interspeech*, 2013, pp. 1751–1755.
24. Dong Yu, Michael L Seltzer, Jinyu Li, Jui-Ting Huang, and Frank Seide, "Feature learning in deep neural networks - a study on speech recognition tasks," in *Proc. of International Conference on Learning Representations*, 2013.
25. Bo Li and Khe Chai Sim, "Noise adaptive front-end normalization based on vector taylor series for deep neural networks in robust speech recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7408–7412.
26. Bo Li, Yu Tsao, and Khe Chai Sim, "An investigation of spectral restoration algorithms for deep neural networks based noise robust speech recognition," in *Proc. of Interspeech*, 2013, pp. 3002–3006.
27. Michael L. Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7398–7402.
28. Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.
29. Andrew L Maas, Quoc V Le, Tyler M O'Neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y Ng, "Recurrent neural networks for noise reduction in robust ASR," in *Proc. of Interspeech*, 2012, pp. 22–25.
30. Xiangtao Meng, Chao Liu, Zhiyong Zhang, and Dong Wang, "Noisy training for deep neural networks," in *Proc. of ChinaSIP 2014*, 2014, pp. 16–20.
31. Guozhong An, "The effects of adding noise during backpropagation training on a generalization performance," *Neural Computation*, vol. 8, no. 3, pp. 643–674, 1996.
32. Yves Grandvalet and Stéphane Canu, "Comments on 'noise injection into inputs in back propagation learning'," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 4, pp. 678–681, 1995.
33. Chris M Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural computation*, vol. 7, no. 1, pp. 108–116, 1995.
34. Yves Grandvalet, Stéphane Canu, and Stéphane Boucheron, "Noise injection: Theoretical prospects," *Neural Computation*, vol. 9, no. 5, pp. 1093–1108, 1997.
35. Jocelyn Sietsma and Robert JF Dow, "Neural net pruning-why and how," in *Proc. of IEEE International Conference on Neural Networks*, 1988, pp. 325–333.
36. Kiyotoshi Matsuoka, "Noise injection into inputs in back-propagation learning," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 3, pp. 436–440, 1992.
37. Russell Reed, RJ Marks, Seho Oh, et al., "Similarities of error regularization, sigmoid gain scaling, target smoothing, and training with jitter," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 529–538, 1995.

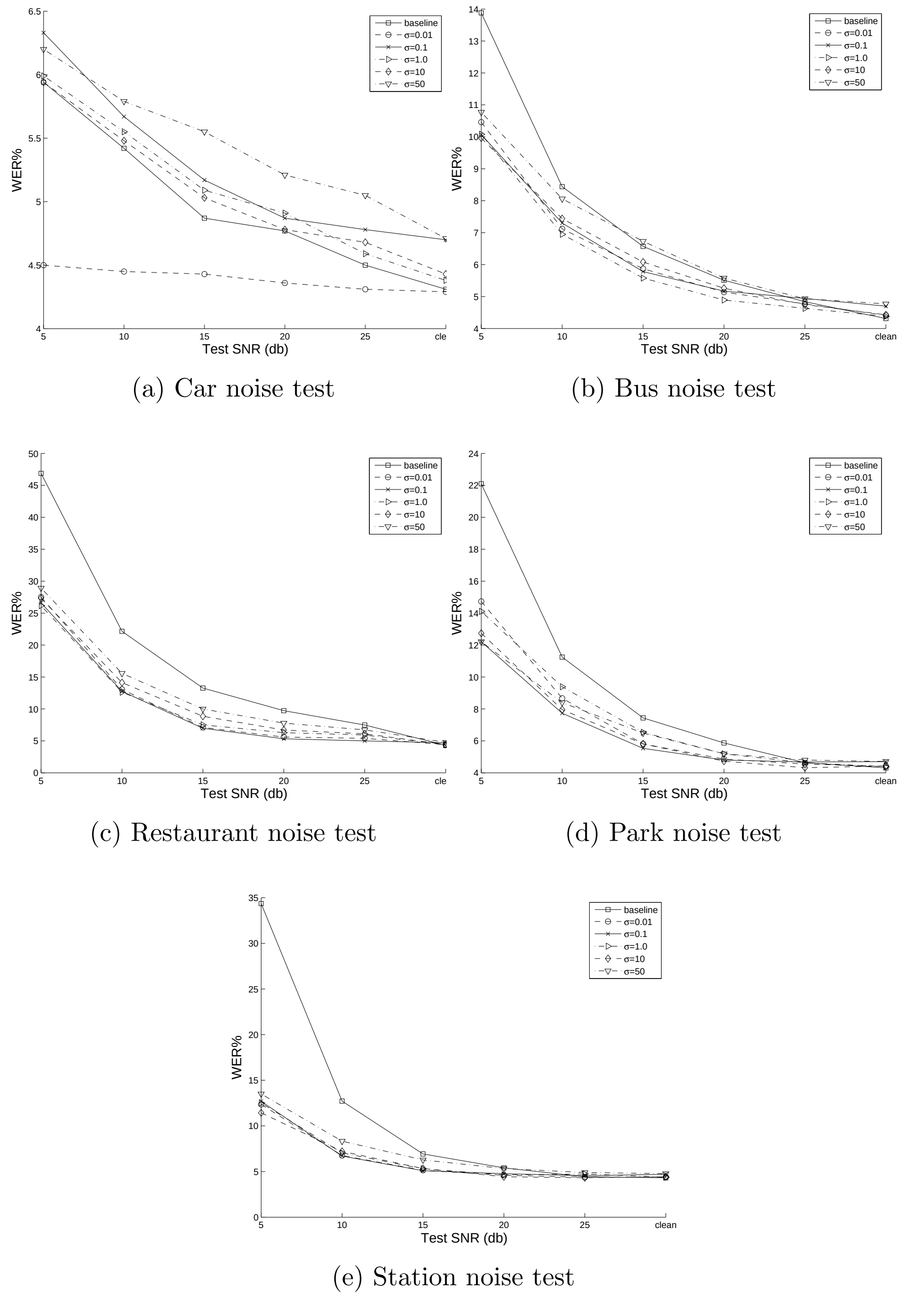


Figure 8 Performance of noise training with different σ_{SNR}. White and Cafeteria noises are injected, and μ_{SNR}=15dB. For each plot, the test data are corrupted by a particular 'unknown' noise. The 'baseline' curves present the results of the system trained with clean speech data, as have been presented in Table 1.