Uncertainty training and decoding methods of deep neural networks based on stochastic representation of enhanced features

Yuuki Tachioka[1], *Shinji Watanabe*[2]

[1]Information Technology R&D Center, Mitsubishi Electric Corporation
[2]Mitsubishi Electric Research Laboratories

Tachioka.Yuki@eb.MitsubishiElectric.co.jp, watanabe@merl.com

Abstract

Speech enhancement is an important front-end technique to improve automatic speech recognition (ASR) in noisy environments. However, the wrong noise suppression of speech enhancement often causes additional distortions in speech signals, which degrades the ASR performance. To compensate the distortions, ASR needs to consider the uncertainty of enhanced features, which can be achieved by using the expectation of ASR decoding/training process with respect to the probabilistic representation of input features. However, unlike the Gaussian mixture model, it is difficult for Deep Neural Network (DNN) to deal with this expectation analytically due to the non-linear activations. This paper proposes efficient Monte-Carlo approximation methods for this expectation calculation to realize DNN based uncertainty decoding and training. It first models the uncertainty of input features with linear interpolation between original and enhanced feature vectors with a random interpolation coefficient. By sampling input features based on this stochastic process in training, DNN can learn to generalize the variations of enhanced features. Our method also samples input features in decoding, and integrates multiple recognition hypotheses obtained from the samples. Experiments on the reverberated noisy speech recognition tasks (the second CHiME and REVERB challenges) show the effectiveness of our techniques.

Index Terms: noise-robust speech recognition, deep neural networks, uncertainty training/decoding, stochastic process of enhanced features

1. Introduction

A deep neural network (DNN) improves the performance of automatic speech recognition (ASR) [1]. We confirmed the effectiveness of DNNs for noisy and reverberant ASR tasks [2, 3]. On the other hand, several methods that were developed for the Gaussian mixture model (GMM) have been applied to DNNs. For example, feature-space maximum-likelihood linear regression (fMLLR), an effective speaker-adaptation technique, is widely used for as the DNN front-end [4]. This paper applies uncertainty techniques to DNNs because uncertainty techniques are successful examples in noisy ASR for GMM-based systems.

In noisy condition, speech enhancement improves the ASR performance, even for a DNN-based systems [5, 6, 7]. However, distortions are consequently introduced to the speech, and this can degrade the ASR performance. This is problematic especially when noise conditions are mismatched between training and decoding time, or when speech enhancement is only applied during decoding, because mismatches of the acoustic model or speech distortion significantly degrades ASR performance.

To address this problem, several methods have been proposed to adjust features according to their reliabilities representing the distortion by speech enhancement. For the GMM, when feature uncertainty can be represented as a Gaussian distribution, the GMM likelihoods are computed based on the expectations with respect to these feature-uncertainty distributions. The expectation is calculated analytically by integrating out marginal parameters, and this marginalization renders models more robust to speech distortions caused by speech enhancement, and it is referred to as the uncertainty-decoding technique. As a result, covariance matrices for the Gaussian distributions of the acoustic models for input features are adjusted corresponding to the extent of uncertainties (i.e., reliability). Many uncertainty methods have been proposed, and their effectiveness for the GMM has been demonstrated experimentally [8, 9, 10, 11, 12, 13, 14]. For example, [10, 11] used a difference vector between noisy and enhanced feature vectors, [12] used a posterior variance of Wiener filters, and [15] used an estimate based on a binary speech/noise predominance model. However, because of an inclusion of non-linear activations in DNNs, it is difficult to handle uncertainty propagations analytically.

This paper proposes uncertainty training and decoding methods for DNNs. Unlike [16], which calculates the expectation operation approximately for the DNN score calculation and for training of DNNs, our method samples some input features based on uncertainties by using the Monte-Carlo method. However, because DNN model training requires considerable computation, efficient sampling is essential. The proposed method focuses on interpolation vectors before and after speech enhancement, and it efficiently represents the feature distributions of enhanced speech vectors by sampling interpolation coefficients probabilistically. In addition, sampling is also performed for decoding, and multiple recognition hypotheses for each sample are combined to further improve the performance.

2. DNN uncertainty training and decoding

The theory behind the uncertainty technique is based on the following conditional expectation operation:

$$\mathbb{E}[f(\boldsymbol{y}_{1:T})|\boldsymbol{x}_{1:T}] \triangleq \int f(\boldsymbol{y}_{1:T})p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T})d\boldsymbol{y}_{1:T}, \quad (1)$$

where $\boldsymbol{x}_{1:T} = \{\boldsymbol{x}_t|t = 1, \ldots, T\}$ is a sequence of T noisy feature vector and $\boldsymbol{y}_{1:T}$ is a sequence of T enhanced features. $f()$ denotes decoding (see Section 2.1) or training (see Section 2.2) depending on the application of our target[1]. $p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T})$ is a

[1] Although $f()$ has several options including an acoustic score function [17], this paper regards $f()$ as an entire decoding process, which returns output sequences.

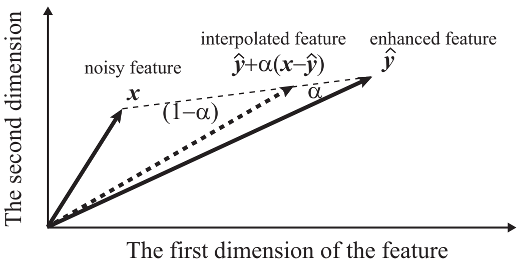

Figure 1: Noisy feature $\boldsymbol{x}$ and enhanced feature $\hat{\boldsymbol{y}}$, and the sampling of feature $\boldsymbol{y}$ based on an interpolation between them.

stochastic representation of an enhanced feature sequence with its uncertainty (see Section 2.3).

2.1. DNN uncertainty decoding

We first focus on uncertainty decoding for DNNs with a hybrid architecture that combines the hidden Markov model (HMM) with the DNN. In this framework, $f()$ in Eq. (1) is represented by the following actual decoding process:

$$\begin{aligned}\hat{W} &= \mathbb{E}\left[\arg\max_{W} p(\boldsymbol{y}_{1:T}|\mathcal{H}_W)p(W)\middle|\, \boldsymbol{x}_{1:T}\right],\\ &= \mathbb{E}\left[W_{\boldsymbol{y}_{1:T}}|\boldsymbol{x}_{1:T}\right],\end{aligned} \tag{2}$$

where W is a word sequence and $\mathcal{H}_W$ is a possible HMM state-sequence given W. $W_{\boldsymbol{y}_{1:T}}$ is a decoded word sequence given input feature sequence $\boldsymbol{y}_{1:T}$. Note that some conventional uncertainty techniques based on the GMM provide an analytical solution to Eq. (2) by integrating out the expectation operations for $\mathbb{E}[p(\boldsymbol{y}_{1:T}|\mathcal{H}_W)|\boldsymbol{x}_{1:T}]$ with a Gaussian-based uncertainty for $p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T})$ (see [17] for more details). However, DNN-based acoustic models cannot obtain such analytical solutions, owing to the presence of nonlinear activation functions; these models require approximations [16, 18].

Rather than using approximations, we adopt a straightforward expectation from Eq. (2), based on a Monte-Carlo sampling, and averaging out multiple outputs at the hypothesis level rather than integrals. These outputs are obtained from decoding processes with different feature samples. The disadvantage to this approach is that it requires the ASR decoding computations for all samples, even though lattice re-scoring can decrease these computations. In addition, it is very difficult to sample $\boldsymbol{y}_{1:T}$ to fully cover a possible input feature space. Instead of directly considering the distribution of sequential input feature $p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T})$, we assume a deterministic relationship for the sampled input feature $\boldsymbol{y}_t$ at the frame t based on a linear interpolation between $\boldsymbol{x}_t$ and $\hat{\boldsymbol{y}}_t$ as:

$$\boldsymbol{y}_t = \hat{\boldsymbol{y}}_t + \alpha(\boldsymbol{x}_t - \hat{\boldsymbol{y}}_t) \text{ for } t = 1, \ldots, T, \tag{3}$$

where α is a linear interpolation coefficient. The geometric meaning of this linear interpolation is shown in Fig. 1. This approach is inspired by uncertainty decoding based on an approximated observation distribution with the covariance matrix obtained by the difference between noisy and enhanced features: $p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T}) \approx \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{y}_t|\hat{\boldsymbol{y}}_t, \left[\alpha(\boldsymbol{x}_t - \hat{\boldsymbol{y}}_t)(\boldsymbol{x}_t - \hat{\boldsymbol{y}}_t)^\top\right])$ in [10, 11]. In fact, Eq. (3) can be regarded as a sigma point for this distribution [19]. Then, we regard the linear interpolation coefficient α as a random variable, and efficiently sample one-dimensional α with a relatively small number of samples.

Thus, our proposed uncertainty decoding with N Monte Carlo samples is represented from Eq. (2) as follows:

$$\begin{aligned}&\hat{W} = R\left[\left\{W_{\boldsymbol{y}^n_{1:T}}\right\}_{n=1}^{N}\right],\\ &\boldsymbol{y}^n_t = \hat{\boldsymbol{y}}_t + \alpha^n(\boldsymbol{x}_t - \hat{\boldsymbol{y}}_t) \text{ for } t = 1, \ldots, T,\ \alpha^n \sim p(\alpha),\end{aligned} \tag{4}$$

where $R[\cdot]$ is performed by using a hypothesis-level integration, e.g., with Recognizer Output Voting Error Reduction (ROVER) [20]. $\alpha^n \sim p(\alpha)$ means that the n-th α is sampled from the distribution $p(\alpha)$. Section 2.3 discusses $p(\alpha)$ in more detail.

2.2. DNN uncertainty training

In a manner similar to the description in Section 2.1, uncertainty training, given a reference word sequence W, can be represented by replacing $f()$ in Eq. (1) with a training procedure:

$$\hat{\Theta} = \mathbb{E}\left[\arg\min_{\Theta} \mathcal{F}_\Theta(\boldsymbol{y}_{1:T}, W)\middle|\, \boldsymbol{x}_{1:T}\right], \tag{5}$$

where $\mathcal{F}_\Theta$ is an objective function of the DNN, e.g., cross entropy (CE) or sequence-discriminative criteria, with the model parameter Θ.

The input features are sampled based on the distribution of a linear interpolation coefficient $p(\alpha)$ similarly to the proposed uncertainty decoding in Section 2.1. Instead of the expectation operation with respect to parameters in Eq. (5), we propose to use a Monte Carlo sampling for an objective function

$$\begin{aligned}\hat{\Theta} &= \arg\min_{\Theta} \mathbb{E}\left[\mathcal{F}_\Theta(\boldsymbol{y}_{1:T}, W)|\, \boldsymbol{x}_{1:T}\right],\\ &\approx \arg\min_{\Theta} \sum_{n=1}^{N} \mathcal{F}_\Theta(\boldsymbol{y}^n_{1:T}, W),\\ &\text{where} \quad \boldsymbol{y}^n_t = \hat{\boldsymbol{y}}_t + \alpha^n(\boldsymbol{x}_t - \hat{\boldsymbol{y}}_t)\forall t, \quad \alpha^n \sim p(\alpha).\end{aligned} \tag{6}$$

For CE training, the objective function with the Monte Carlo sampling is represented as follows:

$$\sum_{n=1}^{N} \mathcal{F}^{\text{CE}}_\Theta(\boldsymbol{y}^n_{1:T}, W) = -\sum_{t=1}^{T}\sum_{n=1}^{N} \log p_\Theta(s_t|\boldsymbol{y}^n_t), \tag{7}$$

where s_t is an HMM state at the frame t, obtained by the Viterbi alignment given W. Thus, the additivity to the objective function enables the expectation operation, simply by using the sampled training data as input features. This approach can also be applied to sequence-discriminative DNN training, e.g., [21]. The proposed approach is motivated by a deep learning method, which has recently been used in the area of image processing [22, 23] to train DNN models by sampling input features based on possible feature changes. Such an approach renders models robust and invariant to these changes.

2.3. Stochastic process for the linear-interpolation coefficient

We sample multiple α's for each utterance by using the following one-dimensional Gaussian mixture with K mixture components to sample α:

$$p(\alpha) = \sum_{k=1}^{K} w_k \mathcal{N}(\alpha|\mu_k, \sigma), \tag{8}$$

where the mean μ_k is empirically determined from some values in $[0, 1]$, so that the input feature $\boldsymbol{y}_t$ is sampled between the noisy feature $\boldsymbol{x}_t$ and the enhanced feature $\hat{\boldsymbol{y}}_t$. The variance σ and the mixture weight $w_k (= 1/K)$ are fixed, and in some experiments $\alpha \in \{\mu_k\}_{k=1}^{K}$ are fixed, i.e., $\sigma \to 0$.

3. Experimental setup

We validated the effectiveness of our proposed approaches with two noisy and reverberated ASR tasks. The first corpus was the second CHiME challenge Track 2 [24], which is a medium-vocabulary task whose speech utterances are taken from the *Wall Street Journal* (*WSJ*) database with non-stationary noise between a -6 and 9 dB signal-to-noise ratio (SNR). The multi-channel non-negative matrix factorization (MNMF) algorithm [25, 26] was used for speech enhancement.

The second corpus was the REVERB challenge [27] simulation data, which is a medium-vocabulary task in reverberant environments, whose utterances are also taken from the *WSJ* [28]. Speech data were created by convolving clean speech with six types of room impulse responses at a distance of 0.5 m (near) or 2 m (far) from the microphones in three rooms (1–3) whose reverberation times were 0.25, 0.5, and 0.75 s, respectively, and by adding relatively stationary noise at 20 dB SNR. Eight microphones were arranged in a circle with a radius of 0.1 m. Multi-channel beamforming with direction-of-arrival estimation and a single-channel dereveberation were applied [3].

The ASR settings were the same for both tasks. Some tuning parameters, e.g., language model weights, were optimized based on the word error rate (WER) of the development set. The vocabulary size was 5k and a trigram language model was used. These systems were constructed using the Kaldi toolkit [29]. Further details are found in [2, 3]. The learning rates were reduced for the proposed uncertainty-training method, because the interpolated training data were similar to the original data and acoustic models tend to be overly tuned. We used 40-dimensional filter bank features with Δ and $\Delta\Delta$. The DNN acoustic models were constructed according to the CE criterion before performing sequential minimum Bayes risk (SMBR) discriminative training [21].

The following six system types were prepared.

1. noisy: decoding $\boldsymbol{x}$ (trained on $\boldsymbol{x}$)
2. enhan (enhanced): decoding $\hat{\boldsymbol{y}}$ (trained on $\boldsymbol{y}$)
3. diff (difference): decoding $\left[\hat{\boldsymbol{y}}^\top, [\boldsymbol{x}-\hat{\boldsymbol{y}}]^\top\right]^\top$
4. uncert(t) (uncertainty training): decoding $\hat{\boldsymbol{y}}$, whereas models were trained on $\hat{\boldsymbol{y}} + \alpha[\boldsymbol{x}-\hat{\boldsymbol{y}}]$ with $\mu_k \in \{0, 0.1, 0.2\}$.
5. uncert(d) (uncertainty decoding): decoding $\hat{\boldsymbol{y}}+\alpha[\boldsymbol{x}-\hat{\boldsymbol{y}}]$ with $\mu_k \in \{0, 0.1, 0.2\}$, whereas models were trained on $\hat{\boldsymbol{y}}$. Their hypotheses were combined using ROVER.
6. uncert(t,d) (combination of uncertainty training and decoding): decoding $\hat{\boldsymbol{y}}+\alpha[\boldsymbol{x}-\hat{\boldsymbol{y}}]$ with $\mu_k \in \{0, 0.1, 0.2\}$, and models were trained with the same features. Their hypotheses were also combined using ROVER.

4. Result and discussion

4.1. The second CHiME challenge: Track 2

Table 1 shows the WER from the second CHiME challenge development set. Speech enhancement by MNMF significantly improved the ASR performance of the DNN system. Concatenating difference features ("diff" in table, this is motivated by [10, 11] but it simply stacks uncertainty observations) to input features reduced the WER for the CE model by 0.23%, and by 0.31% for the SMBR (discriminatively trained) model. This experiment used fixed α's, i.e., $\alpha \in \{0, 0.1, 0.2\}$ ($\sigma \to 0$ in Eq. (8)). The proposed uncertainty decoding ("uncert(d)" in the table) reduced the WER by 0.37% and 0.38% for the CE and SMBR models, respectively. In this case, model re-training was unnecessary but the computational time increased for decoding. The proposed uncertainty training ("uncert(t)") reduced the WER by 0.75% and 0.55% for the CE and SMBR models, respectively. In this case, training time increased, whereas the decoding time was almost the same as it was for "enhan" and "diff". For the DNN acoustic models, it is more effective

Table 1: WER [%] on the development set of the second CHiME challenge (Track 2).

	-6dB	-3dB	0dB	3dB	6dB	9dB	Avg.
*CE							
noisy	51.03	39.59	32.17	26.11	21.71	18.88	31.58
enhan	42.79	33.91	28.71	23.32	20.83	17.76	27.89
diff	43.19	34.21	27.75	23.12	20.30	17.39	27.66
uncert(t)	42.29	32.87	27.63	22.27	20.68	17.10	27.14
uncert(d)	42.19	33.22	28.37	23.38	20.43	17.55	27.52
uncert(t,d)	41.92	32.60	27.48	22.13	20.64	17.02	**26.97**
*SMBR							
noisy	48.05	36.64	29.18	23.60	18.90	17.01	28.90
enhan	39.15	30.95	24.99	20.36	18.54	15.50	24.92
diff	39.42	30.46	24.35	20.56	17.47	15.39	24.61
uncert(t)	37.90	30.64	24.55	20.40	17.57	15.19	24.37
uncert(d)	38.50	30.05	24.58	20.30	18.31	15.49	24.54
uncert(t,d)	37.04	29.72	24.19	19.78	16.98	15.08	**23.80**

Table 2: WER [%] on development set of the second CHiME challenge with the addition of random perturbation to the interpolated points.

σ	-6dB	-3dB	0dB	3dB	6dB	9dB	Avg.
*CE							
uncert(t)							
0	42.29	32.87	27.63	22.27	20.68	17.10	27.14
0.005	41.45	32.13	27.39	22.92	20.15	17.10	26.86
0.010	41.70	32.44	27.51	22.76	20.19	17.30	26.99
0.015	41.08	32.76	27.63	23.01	19.81	16.65	**26.83**
uncert(d)							
0	42.19	33.22	28.37	23.38	20.43	17.55	**27.52**
0.005	42.23	33.19	28.46	23.37	20.42	17.54	27.53
0.010	42.26	33.22	28.53	23.37	20.42	17.58	27.56
0.015	42.26	33.22	28.54	23.34	20.40	17.60	27.56
uncert(t,d)							
0	41.92	32.60	27.48	22.13	20.64	17.02	26.97
0.005	40.85	31.73	27.13	22.82	19.80	16.79	26.52
0.010	40.60	32.04	26.94	22.17	19.59	17.20	26.42
0.015	40.54	31.82	27.36	22.33	19.13	16.36	**26.26**
*SMBR							
uncert(t)							
0	37.90	30.64	24.55	20.40	17.57	15.19	**24.37**
0.005	38.40	30.40	24.86	20.21	18.03	15.34	24.54
0.010	38.72	30.45	25.53	20.73	17.41	15.36	24.70
0.015	38.03	31.02	25.74	21.48	17.85	15.64	24.95
uncert(d)							
0	38.50	30.05	24.58	20.30	18.31	15.49	24.54
0.005	38.44	30.08	24.58	20.31	18.31	15.49	**24.53**
0.010	38.49	29.89	24.71	20.39	18.01	15.70	**24.53**
0.015	38.49	30.20	24.55	20.19	18.29	15.49	**24.53**
uncert(t,d)							
0	37.04	29.72	24.19	19.78	16.98	15.08	**23.80**
0.005	37.72	30.33	24.34	20.08	17.27	15.30	24.17
0.010	37.69	29.84	24.83	20.24	17.01	15.08	24.11
0.015	37.00	30.09	24.84	20.64	17.39	15.50	24.24

Table 3: WER [%] on the evaluation set of the second CHiME challenge, where '+p' refers to the inclusion of random perturbation at $\sigma = 0.015$.

	−6dB	−3dB	0dB	3dB	6dB	9dB	Avg.
*CE							
noisy	44.07	34.56	28.40	20.46	17.13	14.72	26.56
enhan	36.56	27.65	23.50	19.33	16.46	15.04	23.09
diff	38.05	28.58	23.13	18.85	15.62	13.58	22.97
uncert(t)	35.57	27.03	22.57	19.50	15.54	14.18	22.40
+p	35.62	27.29	22.53	18.27	15.77	13.77	22.21
uncert(d)	35.98	27.27	23.31	19.02	15.97	14.59	22.69
+p	35.94	27.26	23.28	18.98	16.01	14.65	22.69
uncert(t,d)	35.23	26.51	22.36	19.15	15.24	14.16	22.11
+p	35.16	26.62	22.42	18.48	15.62	13.49	**21.96**
*SMBR							
noisy	40.91	32.21	26.42	18.64	15.54	13.82	24.59
enhan	32.11	25.22	20.49	16.74	14.46	12.72	20.29
diff	33.44	25.95	20.83	17.04	14.33	12.65	20.70
uncert(t)	32.36	25.82	20.68	17.17	14.16	12.91	20.51
+p	31.40	25.18	20.85	17.58	14.50	12.78	20.38
uncert(d)	31.89	24.64	20.16	16.59	14.22	12.42	19.99
+p	31.85	24.79	20.19	16.50	14.22	12.44	20.00
uncert(t,d)	31.98	24.68	20.31	17.15	13.92	12.57	20.10
+p	30.66	24.73	20.38	17.07	13.97	12.35	**19.86**

to consider uncertainties for training than for decoding. When uncertainties are introduced to both training and decoding ("uncert(t,d)"), the WERs were significantly improved, by 0.92% and 1.12%, for the CE and SMBR models, respectively.

Table 2 shows the effectiveness of random perturbation ($\sigma > 0$ in Eq.(8)) to the interpolated points (see Section 2.3). Although, for all σ's, this method did not improve the ASR performance for uncertainty decoding ("uncert(d)"), it improved the performance for both uncertainty training ("uncert(t)") and the combination of training with decoding ("uncert(t,d)"). In the case of $\sigma = 0.015$, for the CE acoustic model, the WER improved by 0.31% for training, and by 0.71% for the combination of training with decoding. However, this method did not improve the ASR performance for the SMBR model, which is robust to frequent error patterns.

Table 3 shows the WER on the evaluation set, where '+p' denotes the case of $\sigma = 0.015$. In this case, the introduction of uncertainties improved the performance of training more than decoding, and it achieved the best performance in the case of "uncert(t,d)". This trend was similar to that of the development set. In this case, random perturbation to the uncertainty training and both training and decoding improved the performance even for the SMBR model. This shows that perturbation renders the acoustic models more robust to unknown data. Finally, the proposed method reduced the WER from "enhan" for the CE model by 1.13% and for SMBR model by 0.43%, and outperformed the "diff" by 0.12% and −0.41%. These results confirmed the effectiveness of the proposed method.

4.2. The REVERB challenge

Table 4 shows the WER on the development set of the REVERB challenge. The experiments in this section used fixed α's. Although the baseline performance was better than it was with the CHiME challenge, the proposed method was also effective and the trends were similar, i.e., the proposed method was more effective for training than decoding, and the combination further improved the performance.

Table 4: WER [%] on the development set of the REVERB challenge simulation data.

	Room1		Room2		Room3		Avg.
	far	near	far	near	far	near	
*CE							
noisy	6.69	5.16	11.17	7.02	13.18	8.14	8.56
enhan	6.78	5.85	9.86	6.11	10.36	6.97	7.66
diff	6.15	5.01	9.69	6.21	9.82	6.28	7.19
uncert(t)	6.59	5.53	9.29	5.92	9.77	6.13	7.21
uncert(d)	6.74	5.68	9.93	6.11	10.44	6.95	7.64
uncert(t,d)	6.44	5.43	9.17	6.09	9.77	5.98	**7.15**
*SMBR							
noisy	5.36	4.11	9.54	5.52	10.29	6.90	6.95
enhan	5.51	4.57	7.79	5.13	8.21	5.04	6.04
diff	5.29	4.20	7.96	5.20	7.72	5.37	5.96
uncert(t)	5.41	4.30	7.42	5.15	8.11	4.77	5.86
uncert(d)	5.26	4.62	7.59	4.95	8.33	5.46	6.04
uncert(t,d)	5.29	4.18	7.54	5.15	7.86	4.92	**5.82**

Table 5: WER [%] on the evaluation set of the REVERB challenge simulation data.

	Room1		Room2		Room3		Avg.
	far	near	far	near	far	near	
*CE							
noisy	6.44	5.76	11.91	7.46	13.27	8.21	8.84
enhan	6.44	6.05	9.89	6.12	12.04	6.21	7.79
diff	6.18	5.51	9.47	6.16	11.53	7.10	7.66
uncert(t)	6.00	5.69	9.05	5.74	11.17	6.26	7.32
uncert(d)	6.40	5.88	9.89	6.25	12.04	6.28	7.79
uncert(t,d)	5.90	5.62	9.03	5.79	11.05	6.24	**7.27**
*SMBR							
noisy	5.40	5.01	9.64	5.87	10.93	7.20	7.34
enhan	5.73	5.29	7.72	5.35	9.57	5.77	6.57
diff	5.37	4.95	7.83	5.45	9.67	6.19	6.58
uncert(t)	5.40	5.13	7.81	5.58	9.31	6.12	6.56
uncert(d)	5.45	4.98	7.89	5.51	9.40	5.83	6.51
uncert(t,d)	5.25	5.03	7.80	5.42	9.13	5.89	**6.42**

Table 5 shows the WER on the evaluation set. The proposed method improved the WER from "enhan" for CE model by 0.52% and for SMBR model by 0.15%, and outperformed "diff" by 0.13% and −0.01%. Thus, the proposed method improved the ASR performance for two tasks.

5. Conclusions

This paper proposed uncertainty training and decoding methods for DNN acoustic models to address observation uncertainties caused by speech enhancement. Our proposed method did not change the structure or the training and decoding strategy of the DNN. Rather, it realized uncertainty training and decoding with an efficient sampling method for enhanced features. By comparing the introduction of uncertainties to training and decoding, we discovered that the introduction of uncertainty to the training is the most effective. In addition, a random perturbation of interpolated points further improved the performance. The effectiveness of the proposed method was confirmed for noisy and reverberant two ASR tasks. Future work will seek to develop an algorithm that determines the optimal interpolated points depending on the type of noise.

6. Acknowledgements

We would like to thank Dr. Emmanuel Vincent for providing his baseline tools during his stay at MERL.

7. References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal Processing Magazine*, vol. 28, pp. 82–97, 2012.

[2] Y. Tachioka, S. Watanabe, J. Le Roux, and J. Hershey, “Discriminative methods for noise robust speech recognition: A CHiME challenge benchmark,” in *Proceedings of the 2nd CHiME Workshop on Machine Listening in Multisource Environments*, 2013, pp. 19–24.

[3] Y. Tachioka, T. Narita, S. Watanabe, and F. Weninger, “Dual system combination approach for various reverberant environments,” in *Proceedings of REVERB challenge*, 2014, pp. 1–8.

[4] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. Mori, “Linear hidden transformations for adaptation of hybrid ANN/HMM models,” *Speech Communication*, vol. 49, no. 10, pp. 827–835, 2007.

[5] M. Seltzer, D. Yu, and Y. Wang, “An investigation of deep neural networks for noise robust speech recognition,” in *Proceedings of ICASSP*, 2013, pp. 7398–7402.

[6] M. Delcroix, Y. Kubo, T. Nakatani, and A. Nakamura, “Is speech enhancement pre-processing still relevant when using deep neural networks for acoustic modeling?” in *Proceedings of INTERSPEECH*, 2013, pp. 2992–2996.

[7] A. Narayanan and D. Wang, “Investigation of speech separation as a front-end for noise robust speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 826–835, 2014.

[8] J. Arrowood and M. Clements, “Using observation uncertainty in HMM decoding,” in *Proceedings of ICSLP*, 2002.

[9] H. Liao and M. Gales, “Joint uncertainty decoding for noise robust speech recognition,” in *Proceedings of EUROSPEECH*, 2005, pp. 3129–3132.

[10] M. Delcroix, T. Nakatani, and S. Watanabe, “Static and dynamic variance compensation for recognition of reverberant speech with dereverberation preprocessing,” *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 324–334, 2009.

[11] D. Kolossa, R. F. Astudillo, E. Hoffmann, and R. Orglmeister, “Independent component analysis and time-frequency masking for speech recognition in multi-talker conditions,” *EURASIP Journal on Audio, Speech, and Music Processing*, p. ID 651420, 2010.

[12] R. F. Astudillo, *Integration of Short-time Fourier Domain Speech Enhancement and Observation Uncertainty Techniques for Robust Automatic Speech Recognition*. PhD Thesis, Universität Berlin, 2010.

[13] L. Lu, K. Chin, A. Ghoshal, and S. Renals, “Joint uncertainty decoding for noise robust subspace Gaussian mixture models,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 1791–1804, 2013.

[14] D. T. Tran, E. Vincent, and D. Jouvet, “Fusion of multiple uncertainty estimators and propagators for noise robust ASR,” in *Proceedings of ICASSP*, 2014, pp. 5549–5553.

[15] F. Nesta, M. Matassoni, and R. F. Astudillo, “A flexible spatial blind source extraction framework for robust speech recognition in noisy environments,” in *Proceedings of the 2nd CHiME Workshop on Machine Listening in Multisource Environments*, 2013, pp. 33–38.

[16] R. Astudillo and J. da Silva Neto, “Propagation of uncertainty through multilayer perceptrons for robust automatic speech recognition,” in *Proceedings of INTERSPEECH*, 2011.

[17] D. Kolossa and R. Haeb-Umbach, *Robust Speech Recognition of Uncertain or Missing Data - Theory and Applications*. Springer Verlag, 2011.

[18] W. A. Wright, “Bayesian approach to neural-network modeling with input uncertainty,” *IEEE Transactions on Neural Networks,*, vol. 10, no. 6, pp. 1261–1270, 1999.

[19] S. Julier, J. Uhlmann, and H. Durrant-White, “A new method for non-linear transformation of means and covariances in filters and estimators,” *IEEE Transactions on Automatic Control*, vol. 45, pp. 477–482, 2000.

[20] J. Fiscus, “A post-processing system to yield reduced error word rates: Recognizer output voting error reduction (ROVER),” in *Proceedings of ASRU*, 1997, pp. 347–354.

[21] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, “Sequence-discriminative training of deep neural networks,” in *Proceedings of INTERSPEECH*, 2013.

[22] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, “Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3361–3368.

[23] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the Twenty-fifth International Conference on Machine Learning*, 2008, pp. 1096–1103.

[24] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, “The second ‘CHiME’ speech separation and recognition challenge: Datasets, tasks and baselines,” in *Proceedings of ICASSP*, 2013, pp. 126–130.

[25] A. Ozerov, E. Vincent, and F. Bimbot, “A general flexible framework for the handling of prior information in audio source separation,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, pp. 1118–1133, 2012.

[26] H. Sawada, H. Kameoka, S. Araki, and N. Ueda, “Multichannel extensions of non-negative matrix factorization with complex-valued data,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 971–982, 2013.

[27] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Haeb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas, S. Gannot, and B. Raj, “The REVERB Challenge: A common evaluation framework for dereverberation and recognition of reverberant speech,” in *Proceedings of WASPAA*, 2013.

[28] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, “WSJCAM0: a British English speech corpus for large vocabulary continuous speech recognition,” in *Proceedings of ICASSP*, 1995, pp. 81–84.

[29] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, M. Petr, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, “The Kaldi speech recognition toolkit,” in *Proceedings of ASRU*, 2011, pp. 1–4.