UNCERTAINTY ESTIMATION OF DNN CLASSIFIERS

Sri Harish Mallidi[1], *Tetsuji Ogawa*[3], *Hynek Hermansky*[1,2]

[1]Center for Language and Speech Processing, The Johns Hopkins University, Baltimore, U.S.A
[2]Human Language Technology Center of Excellence, The Johns Hopkins University, Baltimore, U.S.A
[3] Department of Computer Science, Waseda University, Tokyo, Japan.

ABSTRACT

New efficient measures for estimating uncertainty of deep neural network (DNN) classifiers are proposed and successfully applied to multistream-based unsupervised adaptation of ASR systems to address uncertainty derived from noise. The proposed measure is the error from associative memory models trained on outputs of a DNN. In the present study, an attempt is made to use autoencoders for remembering the property of data. Another measure proposed is an extension of the M-measure, which computes the divergences of probability estimates spaced at specific time intervals. The extended measure results in an improved reliability by considering the latent information of phoneme duration. Experimental comparisons carried out in a multistream-based ASR paradigm demonstrates that the proposed measures yielded improvements over the multistyle trained system and system selected based on existing measures. Fusion of the proposed measures achieved almost the same performance as the oracle system selection.

Index Terms— Autoencoder, M-delta measure, uncertainty estimation, deep neural networks, multistream ASR

1. INTRODUCTION

Deep neural network (DNN) based speech recognition systems have shown significant improvements over systems based on Gaussian mixture models (GMMs) [1, 2]. The reason for the improvement is attributed to DNNs ability to model complex, non-linear manifolds that may be separating features from different classes of speech sounds[2]. However, if the test data contains variability not seen in the training data, even the best machine learning techniques may fail without much warning. This differentiates machines from higher animals (including humans), who typically know if they are certain about the decisions they are making [3, 4]. Emulating this ability in machines would be desirable.

Uncertainty of the estimates (usually posterior probability estimates of speech classes) could be useful in a number of applications [5, 6, 7, 8, 9]. It can be used in adaptive selection of processing streams, in a multi-stream framework [6, 7]. Uncertainty estimation can also be useful in semi-supervised training, for example in co-training, where reliable estimates of one classifier are used as labels for another diverse classifier, and vice-versa [8, 9]. In this paper we propose two techniques to measure uncertainty of DNN classifiers.

The first measure is based on the following premise: A DNN is best performing and least uncertain about its estimates on the training data. Therefore, uncertainty of a test data can be measured by computing the deviation in probability estimates derived from the test and the training data. We propose to use autoencoders to model DNN outputs. Autoencoders are feed-forward neural networks, used for modeling complex data distributions [10, 11, 12]. First we train autoencoders to reconstruct DNN outputs of training data. Then, reconstruction error on test data is used as measure of DNN uncertainty on the test data.

The second measure is an extension of the earlier proposed M-measure [13]. The M-measure accumulates the divergences of posterior probability vectors, which are spanned in certain time intervals apart. For the short spans, the divergences are small. They increase with increasing time span up to the point where both compared probability vectors come from different coarticulation patterns. Since distortion of a signal could make the probability vectors coming from such large spans more similar, the cumulative divergence curve indicates the quality of probability estimates. The extension of this measure, which was inspired by the segmentation algorithm proposed in [14], compares the difference in divergences coming from the same phoneme as well as different phonemes.

The uncertainty measures proposed are applied for stream-selection in a multi-stream probability estimation, in which different DNNs are trained on specific noise conditions and the DNN having the most acoustic similarity to a given test

This work was supported in parts by the National Science Foundation via award number IIA-0530118, .Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013, by Google via Google faculty award to Hynek Hermansky. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Google, NSF, IARPA, DoD/ARL, or the U.S. Government.

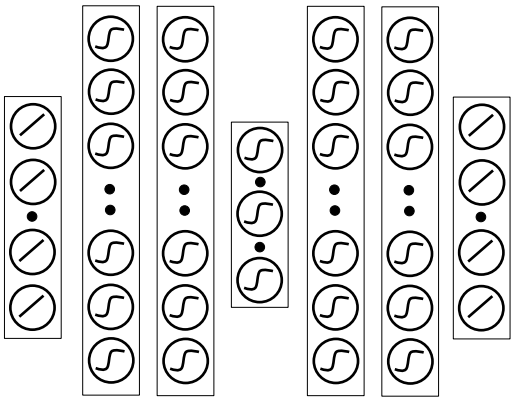

Fig. 1. Illustration of seven layered autoencoder with five non-linear hidden and two linear visible layers.

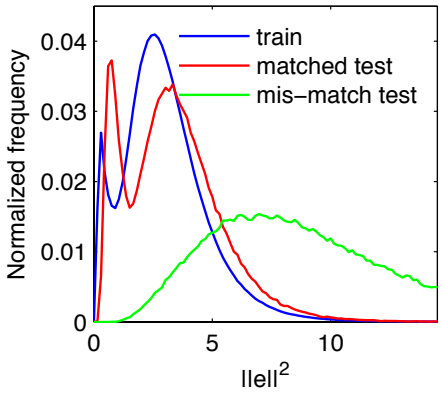

Fig. 2. Illustration of property of autoencoder useful to distinguish matched data and mis-matched data.

utterance is selected. The present study demonstrates that selecting the system trained on matched condition using the proposed measures performs better than the conventional multi-style training approach.

The reminder of the paper is organized as follows: Previously proposed uncertainty measures are described in section 2. Section 3 describes the main principle involved in using autoencoders for uncertainty estimation. M-delta measure is described in section 4. Experimental setups and results are presented in section 5. Section 6 concludes the paper.

2. PAST UNCERTAINTY MEASURES

2.1. Entropy of softmax output

Okawa et. al. and Misra et. al. [6, 7] observed that as noise in test data increases, the output posterior probability distribution from a DNN, trained on clean data, converges to non-informative, uniform distribution. This results in an increase in the entropy of the posterior distribution. Based on this observation, entropy was proposed as a measure of uncertainty.

2.2. M-measure

The M-measure accumulates the divergences of probability estimates spaced over several time-spans. It is defined as

$$\mathcal{M}(\Delta t) = \frac{1}{T-\Delta t}\sum_{t=\Delta t}^{T}\mathcal{D}(\mathbf{p}_{t-\Delta t}, \mathbf{p}_t), \qquad (1)$$

where Δt denotes the time interval between the phoneme posterior probabilities at $t-\Delta t$ and t, $\mathbf{p}_{t-\Delta t}$ and $\mathbf{p}_t$, and $\mathcal{D}(\mathbf{p}, \mathbf{q})$ denotes the symmetric KL divergence between the posteriors,

$$\mathcal{D}(\mathbf{p}, \mathbf{q}) = \sum_{k=0}^{K} p^{(k)} \log \frac{p^{(k)}}{q^{(k)}} + \sum_{k=0}^{K} q^{(k)} \log \frac{q^{(k)}}{p^{(k)}}, \qquad (2)$$

where $p^{(k)}$ denotes the k-th element of a posterior vector $\mathbf{p} \in \mathbb{R}^K$. It has been found that if an ASR system is developed using clean speech, M-measure is higher for clean speech utterances (i.e., known data) and lower for noisy speech utterances (i.e., unknown data). In addition, as the SNR of noisy speech decreases, the M-measure lowers [13]. This means that the M-measure could be effective in determining whether the output of the estimator represents good or bad estimates of speech sounds in speech stream. In multi-stream ASR, the stream (or system) with the highest M-measure can be selected as the most reliable stream (or system) [15] for each utterance.

The M-measures in Eq. (1) are averaged over several time intervals Δt and the result is used as the uncertainty measure,

$$\mathcal{M} = \underset{\{\Delta t\}}{\text{mean}}[\mathcal{M}(\Delta t)], \qquad (3)$$

where $\{\Delta t\}$ consists of 10, 15, 20, $\cdots$, 80 frames (15 intervals).

3. UNCERTAINTY BASED ON AUTOENCODERS

This section describes uncertanty estimation using autoencoders and the details on training of autoencoders.

An autoencoder is a multi-layered feed-forward neural network, used in the context of unsupervised learning. During the training process, parameters of the network are optimized to minimize the squared error cost between an output vector from the autoencoder and the corresponding target vector. The targets used to train the network are inputs themselves. The cost function used to optimize the network parameters ($\mathcal{W}$) is described as

$$\min_{\mathcal{W}} \mathbf{E}||\mathbf{x} - \hat{\mathbf{x}}||_2^2, \qquad (4)$$

where $\mathbf{x}$ is an input vector and $\hat{\mathbf{x}}$ is an output vector from the network. Figure 1 shows the architecture of the autoencoder used. An autoencoder with more than one non-linear

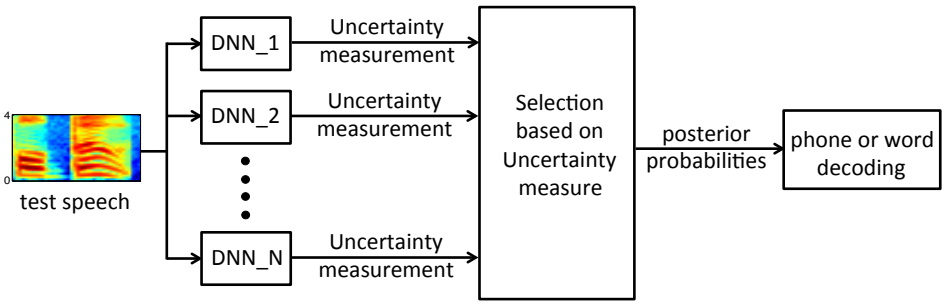

Fig. 3. Stream selection framework to evaluate various uncertainty measures. Each DNN is trained on a specific noise condition.

hidden layer is shown to capture complex, non-linear manifolds present in the training data [11, 17]. In order to avoid a trivial identity mapping (the network weights equal to the unit matrix), the number of nodes in the third hidden layer are chosen to be fewer than the input (and output) layer.

Since the network is trained to minimize the reconstruction error, a vector sampled from the distribution of training data will yield a low reconstruction error compared to vectors drawn from a different distribution. This property is illustrated in Fig. 2, which shows distributions of l_2 norm of reconstruction error vectors ($||e||^2$), computed from the training data (train), data similar to the training data (matched test), and data that deviate from the training data (mis-match test). Figure 2 illustrates that the reconstruction error is a good indicator for measuring the mismatch between the training and test data.

Mallidi et al. [18] first proposed to use the reconstruction errors from autoencoders to measure uncertainty of DNN classifiers. Autoencoders used in [18] are five layered neural networks with three hidden layers. The feature representation used by [18] to train autoencoders is multi-class linear discriminant analysis (LDA) transformed pre-softmax outputs. Context dependent states were used as classes for estimation of LDA transformation. The hidden layers consist of neurons with tanh nonlinearity. Autoencoders are trained using a mean squared error cost function (Eq. 4), starting with random initialization. We have observed that the architecture used in [18] is not robust against the choice of the input dimensionality and number of layers. We attempt to make autoencoders robust against the choice of such hyper parameters by switching to sigmoidal nonlinearity, and initializing networks with restricted Boltzmann machine (RBM) pretraining [22].

4. M-DELTA MEASURE

The original M-measure assumes that the distance between probability estimates over several time-spans should be large for known data. However, this is not always accurate. If two posteriors are from the same phoneme class, the distance between them should be small, irrespective of the time intervals. This means that the original M-measure ignores the effect of the posterior pairs that are separated by large time intervals but belong to the same phoneme class. It accumulates a symmetric KL divergence between the posteriors without considering this kind of phoneme dependency.

More formally, we introduce the idea of within-class and across-class M-measures, $\mathcal{M}^{\mathrm{wc}}$ and $\mathcal{M}^{\mathrm{ac}}$, to represent the accumulated KL-divergence computed from a data pair from the same phoneme class and that from a data pair from different classes, respectively. The new M-delta measure is defined using these within- and across-class M-measures as

$$\mathcal{M}delta = \mathcal{M}^{\mathrm{ac}} - \mathcal{M}^{\mathrm{wc}}. \tag{5}$$

We assume that the M-measure can be decomposed into

$$\mathcal{M}(\Delta t) = p^{\mathrm{wc}}(\Delta t) \cdot \mathcal{M}^{\mathrm{wc}} + p^{\mathrm{ac}}(\Delta t) \cdot \mathcal{M}^{\mathrm{ac}} + \epsilon_{\Delta t}, \tag{6}$$

where $\mathcal{M}(\Delta t)$ denotes the original M-measure defined using Eq. (1), which is determined for each utterance; $p^{\mathrm{wc}}(\Delta t)$ and $p^{\mathrm{ac}}(\Delta t)$ denote the probability of a pair of frames separated by Δt being instances from the same and different phonemes, respectively; and $\mathcal{M}^{\mathrm{wc}}$ and $\mathcal{M}^{\mathrm{ac}}$, the within- and across-class M-measures being estimated for each utterance. $p^{\mathrm{wc}}(\Delta t)$ and $p^{\mathrm{ac}}(\Delta t)$ are determined from the training data transcriptions.

The error term $\epsilon_{\Delta t}$ is included because Eq. (6) is an approximate representation of the M-measure. Although $p^{\mathrm{wc}}(\Delta t)$ and $p^{\mathrm{ac}}(\Delta t)$, which are computed from the training data, are reliably estimated, these probabilities actually differ from those computed from the test utterances, because the variety of phonemes in a test utterance is limited. The redefined M-measure described using Eq. (6) can be written redundantly with several Δt values to minimize the overall error of the within- and across-class M-measures. Assume that $\mathbf{y}$, $\mathbf{A}$, $\mathbf{x}$, and ϵ are given as

$$\mathbf{y} = \left[\begin{array}{ccc} \mathcal{M}(\Delta t_1) & \cdots & \mathcal{M}(\Delta t_N) \end{array} \right]^{\mathrm{T}} \in \mathbb{R}^N \tag{7}$$

$$\mathbf{A} = \left[\begin{array}{cc} p^{\mathrm{wc}}(\Delta t_1) & p^{\mathrm{ac}}(\Delta t_1) \\ \cdots & \cdots \\ p^{\mathrm{wc}}(\Delta t_N) & p^{\mathrm{ac}}(\Delta t_N) \end{array} \right] \in \mathbb{R}^{N \times 2} \tag{8}$$

$$\mathbf{x} = \left[\begin{array}{cc} \mathcal{M}^{\mathrm{wc}} & \mathcal{M}^{\mathrm{ac}} \end{array} \right]^{\mathrm{T}} \in \mathbb{R}^2 \tag{9}$$

$$\epsilon = \left[\begin{array}{ccc} \epsilon_{t_1} & \cdots & \epsilon_{t_N} \end{array} \right]^{\mathrm{T}} \in \mathbb{R}^N \tag{10}$$

Then, Eq. (6) can be written as

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \epsilon. \tag{11}$$

In this case, the within- and across-class M-measures can be estimated as a least square solution:

$$\mathbf{x} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{y}. \tag{12}$$

The experiments below used the values $(\Delta t_1, \Delta t_2, \cdots, \Delta t_N)$ = (1, 2, 3, 4, 5, 10, 15, 20, $\cdots$, 75, 80) and $N = 20$, which were determined by conducting preliminary experiments. Note that higher M-delta values indicate more reliable probability estimates.

5. PHONEME RECOGNITION EXPERIMENT

Experimental comparisons were made in a stream selection framework on TIMIT and Aurora4 databases.

5.1. Stream selection framework

The stream selection framework used is shown in Fig. 3. This framework contains several DNN-based classifiers in parallel. Each DNN classifier is referred to as "stream." A sequence of posterior probability vectors is computed for each stream by forward passing a given test utterance through the corresponding DNN. Posteriors of the least uncertain stream are selected, and provided as an input to a phoneme decoder.

In each stream, the DNN is trained on a specific noise condition. This results in a multi-stream framework where each stream is really good in a specific noise condition. For a given test utterance, selecting posterior estimates from the stream having the most similar acoustic property as the test utterance, results in the lowest error rate. Therefore, we use the phoneme error rates of the stream selection framework to evaluate various uncertainty measures.

5.2. Stream selection experiments on TIMIT

5.2.1. Experimental setup

We used the TIMIT speech dataset for the stream selection experiments [19]. The training set contains 3696 SI and SX utterances from 462 speakers. This totals to 3.12 hours of speech. These are clean, read speech sentences. We used the core development set for the purpose of testing. Five versions of original training set are created by corrupting the clean training speech with four types of additive noise, at various signal-to-noise ratios ranging from 0 dB to 20 dB. We used car, babble, buccaneer1, and buccaneer2 noises from NOISEX database [20]. The original clean training set and four noisy training sets are combined to create a multi-condition training set. The six versions (one clean + four noisy + one multi-condition) of training sets are used to train six different DNNs, where five of them are trained on a specific acoustic condition, and one DNN is trained on multi-condition data. We used a depth of six hidden layers, and each hidden layer consist of 1024 sigmoidal units. Similar to previous studies [21], DNNs are trained on 40 dimensional Mel filter-bank energy features. The DNNs are pre-trained using RBM [22] and fine-tuned using the cross-entropy cost function. The targets used for fine-tuning are context dependent triphone states, generated using a GMM/HMM system trained on clean MFCC features.

We used the development set for testing the models. The development set consists of 34 minutes of speech. Similar to the training set, we corrupted the development set with car, babble, buccaneer1, buccaneer2, destroyerops, exhibition hall, f16 and factory noises, at signal-to-noise ratios of 0, 5, 10, 15 and 20 dB. Four types of noise in this set are seen acoustic variability and the other four noises are unseen acoustic variability in the training set. The whole development set (clean and noisy versions) is referred to as test set from here on.

5.2.2. Experimental result

Table 1 shows the results of test set in various streams.

For the purpose of showing the upper limit of performance, the **oracle** selection technique is defined as selecting the stream which has the most similar acoustic condition of given test data. In the present study, we used two types of oracle stream selection techniques as follows:

- **Utterance oracle**: We select a stream with the lowest error rate for each utterance.
- **Matched condition**: We select a stream trained on the same noise for test utterance with seen noise. Whereas for test utterance with unseen noise, we select a stream trained with multi-condition data.

We can infer that error rates of the condition-level oracle streams are always less than those of individual streams (i.e., clean, car, babble, buccaneer1, and buccaneer2). In addition, the utterance-level oracle streams performs better than the condition-level oracle streams.

Uncertainty measures used for stream selection are as follows:

- **Entropy**: Stream selection based on entropy minimization
- **M**: Stream selection based on M value maximization
- **M-delta**: Stream selection based on M-delta maximization
- **AE_LDA**: Stream selection based on minimization of reconstruction errors from autoencoder

Table 1. Comparison of various uncertainty measures using stream-selection results on TIMIT database.

	seen noises					unseen noises				
Train \Test	clean	car	babble	bucc1.	bucc2.	destops.	exhall	f16	factory	Avg. PER (%)
clean	20.9	34.2	58.3	65.4	65.0	59.2	56.8	62.6	61.6	53.8
car	23.8	22.8	58.1	65.2	64.6	56.1	54.6	62.7	60.6	52.1
babble	30.8	33.1	37.5	38.1	44.6	50.6	53.0	42.0	48.6	41.2
bucc1.	35.4	41.3	53.7	38.1	44.9	50.6	53.0	42.0	48.6	45.3
bucc2.	37.0	45.4	58.3	45.0	37.6	50.7	56.3	46.0	51.7	47.6
Multi-condition	22.2	24.9	39.4	42.0	43.0	39.7	38.4	39.6	40.8	36.7
Utterance Oracle	18.4	20.5	34.7	34.5	34.8	37.0	34.8	35.3	38.2	32.0
Matched condition	20.9	22.8	37.5	38.1	37.6	39.7	38.4	39.6	40.8	35.0
Entropy	22.0	24.8	40.9	43.2	48.5	44.3	39.7	40.5	42.6	38.5
M measure	22.1	24.8	40.8	38.7	41.2	40.8	39.6	39.2	41.8	36.6
M-delta measure	22.1	24.7	40.0	38.3	41.1	41.0	39.2	39.0	41.6	36.3
AE_LDA	20.9	22.9	37.0	37.2	37.1	41.0	37.8	39.0	42.0	35.0
AE_LDA+M-delta	20.9	22.9	36.8	36.6	36.8	39.8	37.2	39.0	41.0	34.6

- **AE_LDA + M-delta**: Stream selection based on combination of reconstruction error minimization and M-delta maximization. Each measure is normalized across streams and the sum of normalized measures is used for stream selection.

It is evident from Table 1 that in seen noises, streams trained on a matched noise condition, which corresponds to the condition-level oracle stream, perform better than the streams trained on multi-condition data. Whereas, in the case of unseen noises, choosing the multi-condition stream performs better, as it is more generalizable to unseen noises than condition specific streams.

Table 1 shows that entropy of posterior probability, obtained at the output of DNN can be erroneous. M measure is performing better than entropy, which suggests rather than looking at a single frame, measures which look at temporal dynamics of posteriors are better. Further improvement to M measure is obtained by using M-delta measure.

Similar to [18], in each stream, we used LDA transformed pre-softmax outputs of the DNN to train an autoencoder corresponding to that stream. Stream selection based on the autoencoders is referred as AE_LDA. From Table 1, it is evident that AE_LDA is performing better than all the other measures. Also, the performance of AE_LDA is matching with the condition-level oracle stream. This shows that, in seen noisy cases, AE_LDA is successfully selecting condition specific streams and in unseen noisy cases, it is selecting the multi-condition stream. The reason for the better performance could be the ability of autoencoders to model distributions lying on a complex non-linear manifolds [11]. In addition, combination of AE_LDA and M-delta seems to improve over the condition-level oracle stream. This implies that the AE_LDA+M-delta is able to select a stream, not just based on similarity with the stream's training data, but also based on the stream's performance.

5.3. Stream selection experiment on Aurora4

We present stream selection experiments performed using Aurora4 database. In this experiment, we attmpt to demonstrate that the effectiveness of the proposed measures on TIMIT is generalizable. The Aurora4 task is a small scale (14 hour), medium vocabulary speech recognition task, aimed at improving noise and channel robustness [23]. Aurora4 database is based on the DARPA Wall Street Journal (WSJ0) corpus which consists of clean recordings of read speech, with 5000 word vocabulary size. The training set consists of 14 hours of clean speech, from 83 speakers, sampled at 16 kHz. The original Aurora4 test set contains simultaneous recordings in 14 different acoustic conditions, but for this study we used only clean subset of it. The clean subset of Aurora4 test set is referred to as test set. Similar to the stream selection setups in TIMIT database, we created five versions of training and test set. We used car, babble, buccaneer1, and buccaneer2 noises from NOISEX database [20]. Table 2 shows the stream selection results on the Aurora4 database. From this table, we can conclude that the proposed measures provide significant improvements over the conventional measures. These results also indicate that the proposed measures are generalizable to other databases.

6. CONCLUSION

Two new measures for uncertainty estimation in DNN-based classifiers were proposed. Experimental comparisons carried out in a multi-stream phoneme recognition paradigm demonstrated the effectiveness of the proposed measures. The proposed measures yielded improvements over the existing measures, and achieved almost the same performance as the oracle performance. In addition, the stream selection framework with proposed uncertainty estimation performed more robust against noise than the conventional multi-condition

Table 2. Comparison of various uncertainty measures using stream-selection results on Aurora4 database.

	seen noises					unseen noises				
Train \Test	clean	car	babble	bucc1.	bucc2.	destops.	exhall	f16	factory	Avg. WER (%)
clean	6.5	18.8	76.0	76.1	79.7	59.6	68.6	62.5	73.8	58.0
car	7.7	7.0	65.4	66.9	74.2	51.3	58.8	57.3	69.2	50.9
babble	14.6	24.3	22.2	49.5	60.8	38.1	24.5	32.3	35.8	33.6
bucc1.	19.8	38.1	64.2	24.0	37.3	45.0	58.6	26.3	44.9	39.8
bucc2.	18.5	45.5	76.5	34.6	22.4	42.1	65.5	32.6	51.6	43.3
Multi-condition	8.7	12.5	32.7	43.3	50.0	37.7	32.1	33.3	40.7	32.4
Utterance Oracle	4.3	5.3	20.4	21.1	20.5	27.4	21.6	19.7	30.1	18.9
Matched condition	6.5	7.0	22.2	24.0	22.4	37.7	32.1	33.3	40.7	25.1
Entropy	8.2	12.0	36.3	41.3	48.1	39.7	34.5	33.6	43.0	33.0
M measure	6.7	8.9	38.2	30.2	34.2	41.8	35.8	30.2	46.8	30.3
M-delta measure	6.7	8.3	30.5	26.4	30.4	40.5	31.2	28.4	44.8	27.5
AE_LDA	6.7	7.0	22.5	23.9	23.5	37.3	24.8	25.1	39.9	23.4

training approach. The measures were shown to generalize well to multi-stream LVCSR system developed on AURORA4 database.

7. REFERENCES

[1] F. Seide et al., "Feature engineering in context-dependent deep neural networks for conversational speech transcription," *in Proc. ASRU*, Dec. 2011, pp. 24-29.

[2] G. Hinton et al., "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82-97, Nov. 2012.

[3] M. Sheffers and M. Coles, "Performance monitoring in confusing word: Error brain activity, judgments of response accuracy, and types of errors," *J. Exp. Psychol.*, vol. 26, no. 1, pp. 141-151, 2000.

[4] J. Smith and D. A. Wahsburn, "Uncertainty monitoring and metacognition by animals," *Current Directions Psychol. Sci.*, vol. 14, no. 1, pp. 19-24, 2005.

[5] J. Kittler and M. Hatef, "On combining classifiers," *IEEE Trans. Pattern Anal. & Machine Intel.*, vol. 20, pp. 226-239, 1998.

[6] S. Okawa et al., "Multi-band speech recognition in noisy environments," *in Proc. ICASSP*, 1998, pp.641-644.

[7] H. Misra et al., "New entropy based multi-stream combination," *in Proc. ICASSP*, 2004, vol.1, pp. 741-744.

[8] S. Ganapathy et al., "Noisy channel adaptation in language identification," *in Proc. SLT*, Dec. 2012, pp.307-312.

[9] S. Ganapathy et al., "Unsupervised channel adaptation for language identification using co-training," *in Proc. ICASSP*, May 2013, pp. 6857-6861.

[10] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks." *AIChE Journal*, vol. 37, no. 2, pp. 233-243, 1991.

[11] B. Yegnarayana and S. Kishore, "AANN: an alternative to GMM for pattern recognition," *Neural Networks*, pp.459-469, 2002.

[12] T. N. Sainath et al., "Auto-encoder bottleneck features using deep belief networks," *in Proc. ICASSP*, March 2012, pp. 4153-5156.

[13] H. Hermansky et al., "Mean temporal distance: Predicting ASR error fromtemporal properties of speech signal," in Proc. ICASSP, 2013, pp. 7423-7426.

[14] J. Cohen, "Segmenting speech using dynamic programming," *J. Acoust. Soc. Amer*, vol. 69, no. 5, pp. 1430-1438, 1981.

[15] E. Variani et al., "Multi-stream recognition of noisy speech with performance monitoring," *in Proc. INTERSPEECH*, Aug. 2013, pp. 2978-2981.

[16] T. Ogawa et al., "Stream selection and integration in multi-stream ASR using GMM-based performance monitoring," *in Proc. INTERSPEECH*, Aug. 2013, pp. 3332-3336.

[17] C. M. Bishop, "Pattern recognition and machine learning," Springer, 2006.

[18] Sri H. Mallidi et al., "Autoencoder based multi-stream combination for noise robust speech recognition," *in Proc. INTERSPEECH*, Sept. 2015.

[19] J. S. Garofolo et al., "Timit acoustic-phonetic continuous speech corpus," *Linguistic Data Consortium*, Philadelphia, 1993.

[20] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Commun.*, vol. 12, no. 3, pp. 247-251, 1993.

[21] A. Mohamed et al., "Understanding how deep belief networks perform acoustic modelling," *in Proc. ICASSP*, 2012, pp. 4273-4276.

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

[23] N. Parihar and J. Picone, "Aurora working group: DSR front end LVCSR Evaluation," *Technical Report*, 2002.

[24] Y. Li et al., "Incremental online feature space MLLR adaptation for telephony speech recognition," *in Proc. ICSLP*, 2002, pp. 1417-1420.