Available online at www.sciencedirect.com

ScienceDirect

Computer Speech & Language 46 (2017) 481–495

www.elsevier.com/locate/csl

Multi-style learning with denoising autoencoders for acoustic modeling in the internet of things (IoT)☆

Payton Lin[a], Dau-Cheng Lyu[b], Fei Chen[c], Syu-Siang Wang[a], Yu Tsao[a,*]

[a] *Research Center for Information Technology Innovation, Academia Sinica, Section 2, Academia Road, Nankang District, Taipei 11529, Taiwan*
[b] *ASUSTeK Computer Inc., Taipei, Taiwan*
[c] *Department of Electrical and Electronic Engineering, Southern University of Science and Technology, Shenzhen, China*

Received 18 April 2016; received in revised form 13 January 2017; accepted 3 February 2017
Available online 22 February 2017

Abstract

We propose a *multi-style learning* (*multi-style* training + deep *learning*) procedure that relies on deep denoising autoencoders (DAEs) to extract and organize the most discriminative information in a training database. Traditionally, multi-style training procedures require either collecting or artificially creating data samples (e.g., by noise injection or data combination) and training a deep neural network (DNN) with all of these different conditions. To expand the applicability of deep learning, the present study instead adopts a DAE to augment the original training set. First, a DAE is utilized to synthesize data that captures useful structure in the input distribution. Next, this synthetic data is combined and mixed within the original training set to exploit the powerful capabilities of DNN classifiers to learn the complex decision boundaries in heterogeneous conditions. By assigning a DAE to synthesize additional examples of representative variations, *multi-style learning* makes class boundaries less sensitive to corruptions by enforcing back-end DNNs to emphasize on the most discriminative patterns. Moreover, this deep learning technique mitigates the cost and time of data collection and is easy to incorporate into the internet of things (IoT). Results showed these data-mixed DNNs provided consistent performance improvements without even requiring any preprocessing on the test sets.

Keywords: Deep learning; Deep neural networks; Multi-style train"ing; Deep denoising autoencoders; Mixed training; Representation learning; Data combination; Data synthesis; Noise injection theory; Feature compensation; Automatic speech recognition; Internet of things (IoT)

1. Introduction

Voice access to services is an essential component of the user interface (Rabiner, 2003). For instance, internet of things (IoT) systems use voice and data to connect objects to offer new services around people (Chaouchi, 2013) and to provide context-aware information about human behavior and the environment (Higuchi et al., 2015; Feng et al., 2015). Humans are often the integral part of the IoT system (Stankovic, 2014) so consumer-centric applications must deliver robustness to real-world noises (Li et al., 2015), reverberation (Hsiao et al., 2015), and different speaking styles (Badino et al., 2016). Due to the growing popularity of low-bandwidth miniature devices, voice interfaces must also maintain performance while subjected to compression, packet loss, and artifacts from wireless

☆ This paper has been recommended for acceptance by Roger K. Moore.
* Corresponding author.
E-mail address: yu.tsao@citi.sinica.edu.tw (Y. Tsao).

http://dx.doi.org/10.1016/j.csl.2017.02.001
0885-2308/

communication systems (Parihar and Picone, 2002). Since speech has to be recognized in a wider variety of conditions than ever before (Virtanen et al., 2012), the ultimate challenge (Deng and Huang, 2004) of reducing machine error rates under all deployment environments remains a priority in IoT paradigms due to the pervasive presence of tags and sensors (Atzori et al., 2010). For instance, the ability to handle all types of noise, channel, and bandwidth mismatches would have practical importance for users of older devices or Bluetooth microphones (Li et al., 2012) and for obtaining training data in resource-poor languages (Seltzer and Acero, 2007).

Recent advances in deep learning (Hinton et al., 2012) have led to breakthroughs in accuracy for automatic speech recognition (ASR). Deep neural networks (DNNs) generalize to similar patterns and can infer the most discriminative parts of each feature by training models with several representational levels to optimize a final objective. As long as the training data is sufficiently representative, DNNs suppress small perturbations in the input features as the internal representations move to higher layers (Yu and Deng, 2015; Yu et al., 2013). However, DNNs tend to be sensitive to changes and over-fitted when training data is either limited (Qi et al., 2013) or if the test samples deviate significantly from the training samples (Fujita et al., 2015). One straightforward approach to incorporating robustness into DNNs is *multi-style training* (Lippmann et al., 1987) from data that is collected in a variety of conditions (Seltzer et al., 2013). DNN multi-style training was defined in Weng et al., (2015) as: either collecting or artificially creating (e.g., by corrupting the clean database with noise samples of various levels and types) acoustic samples under different acoustical environments and training DNNs with all these data. The benefits of training DNNs with multi-style training data is supported by the following experiments and conclusions:

1) Training a DNN on mixed data reduces the average Euclidean distance and variance for output vectors at each hidden layer, indicating that irrelevant variations in the input are considered (Li et al., 2012).
2) Injecting noises during model training makes the class boundaries less sensitive to corruptions by enforcing DNNs to emphasize on the most discriminative patterns of the signal (Yin et al., 2015).
3) Augmenting the original training set with additional data can improve generalization by expanding the training material (Kämmerer and Küpper, 1990), blurring idiosyncrasies (Elman and Zipser, 1988), reducing sensitivity (Matsuoka, 1992), penalizing irrelevant variables (Granvalet, 2000), using more of the units that contribute independently to the solution (Sietsma and Dow, 1991), improving fault tolerance (Reed et al., 1995), providing a form of regularization (Bishop, 1995), and by implicitly altering the training objective function (An, 1996).

Previous multi-style techniques (e.g., training DNNs with mixed band-width data (Li et al., 2012), mixed speech data (Weng et al., 2015), or noise-injected data (Yin et al., 2015)) have provided higher recognition accuracy by either intentionally or randomly transforming the training set. However, one disadvantage of generating artificial data is that the only parameter that effectively controls the amount of regularization is the magnitude of the distortions (Simard et al., 1991). For instance, the noise injection procedure will generate an over-smooth curve and a biased approximation if the statistical property of the noise defines too large a neighborhood around each input (Seghouane et al., 2002). For these reasons, the aim of deep learning research (Bengio et al., 2013) is to make learning algorithms less dependent on this type of prior knowledge and feature engineering to reduce the time and effort that goes into the design of data transformations and preprocessing pipelines. For example, front-end DNNs and deep denoising autoencoders (DAEs) (Vincent et al., 2010) have been proposed to reconstruct clean data from noisy data before combining with a back-end DNN (Du et al., 2014; Mimura et al., 2015). In fact, DAEs (Xie et al., 2012) are related to the DNN noise injection theory (Yin et al., 2015) when noises are randomly selected and intentionally injected to the original data prior to being fed to a denoising function where the target outputs are the original data. However, the DAE approach differs from the artificial synthesis of noise injected data since DAEs exploit many layers of nonlinear transformations to learn more efficient mapping functions and to discover more abstract concepts via multiple levels of distributed representations (Lu et al., 2014; Feng et al., 2014). Therefore, the present study explores how deep learning techniques such as DAE-based feature mapping can be further incorporated into the procedural steps of synthesizing a multi-style training set.

By convention, previous research using front-end DAEs have mostly followed the training procedure outlined in (Seltzer et al., 2013) of training with enhanced features. By processing both the training and testing data with the

same algorithm, any consistent errors or artifacts introduced by the front-end can be learned by the classifier. However, an imperfect enhancement process may discard useful information (Li and Sim, 2013). In addition, front-ends invariably introduce distortions which can negatively affect the performance (Narayanan and Wang, 2014). Although previous studies (Gao et al., 2015; Qian et al., 2015) have reported gains using synthesized training data containing multiple conditions, it has been conjectured that enhancing the features during multi-style training may cause the DNN to be less robust to mismatched conditions (e.g., SNR or channel variations) as the network sees fewer variations in the data (Seltzer et al., 2013). Therefore, we instead propose to mix the DAE-synthesized training data with the original training data to take advantage of the powerful discriminative abilities of DNN classifiers to learn complex decision boundaries in heterogeneous conditions. This mixed treatment of original data and DAE-synthesized data has never been analyzed before and can be considered as a deep learning procedure within the data combination paradigm (Deng and Li, 2013). The first step involves multi-style training except we instead use a DAE (Lu et al., 2013) to collect and create the new samples by transforming source data into target data. In the model-space, a data-mixed DNN is trained with both the original and the DAE-synthesized data to extract useful information for distinguishing between different class labels while suppressing irrelevant variations at the output layers. By generating a mixed training set and fine-tuning back-end DNNs with DAE-synthesized examples of representative variation, this "*multi-style learning*" (*multi-style* training + deep *learning*) procedure expands the scope of deep learning (Bengio et al., 2013; Bengio and Lee, 2015) into multi-style training.

In this paper, the multi-style learning procedure will be evaluated on tasks that are critical in the IoT:

(1) Noise robustness **(2)** Bandwidth extension **(3)** Channel mismatch.

Incorporating invariance into a unified system will be especially useful for hardware industries, as this would mitigate the cost and time of data collection (e.g., by allowing data to be recorded simultaneously from a variety of devices). Since practical situations often require using only one training set to operate at different SNR (Webb, 1994), the training sets for multi-style learning will be synthesized by simultaneously mixing original data and DAE-synthesized data after the data combination stage. This procedure randomly mixes training data at the mini-batch level as in Gao et al., (2016), which differs from the sequential mixed-transfer of multi-style data (You and Xu, 2014). Training sets for multi-style learning will be evaluated on the original test sets to determine if invariance is achieved by adding examples of inputs transformed under the desired invariance group while maintaining the same targets for the raw data (Leen, 1995). In this preliminary study, multi-style learning sets will be compared with original and DAE-synthesized training sets (for DAE frameworks that require matched preprocessing on the test sets).

The rest of this paper is organized as follows: Section 2 reviews related work on DAE-based data synthesis. Section 3 proposes the multi-style learning procedure. Section 4 presents experimental ASR setups for three IoT tasks including the Aurora-4 noise robustness task, a Mandarin broadcast news corpus, and a dataset prepared by ASUSTeK Computer Inc. Section 5 discusses the results. Section 6 concludes.

2. DAE-based data synthesis

The goal of feature compensation is to estimate a mapping function to transform a set of features to a reference one (e.g., Lee, 1998; Tsao et al., 2014; Wang et al., 2014). These techniques form a parametric function to characterize the transformation from source features and target features, where the parameters of the functions are estimated based on some optimality criterion. The source features are usually lower-quality (e.g., speech features that are recorded from adverse acoustic conditions), while the target features are usually high-quality (e.g., speech features that are recorded from clean acoustic conditions). Feature mapping techniques are used to estimate alternative feature representations for training and decoding, and can substantially reduce the interference and noise. For instance, front-end DNNs and front-end DAEs have been used to reconstruct a clean spectrum or a feature mapping across different feature domains (Ishii et al., 2013; Feng et al., 2014; Xu et al., 2014; Ma et al., 2015; Hsiao et al., 2015). These deep learning processes take advantage of multiple layers of nonlinear processing units to learn the high-order statistical information in the data, as shown in the DAE architecture in Fig. 1. Conventionally, the front-end preprocessing is applied to enhance both the training data and the test data to ensure matched features (Han et al., 2015). For further improvements, the front-end preprocessing can also be combined with post-processing technology such as temporal

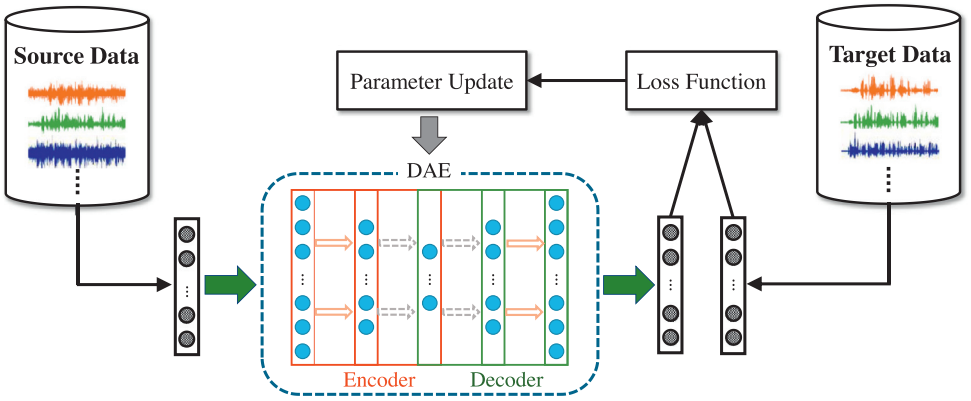

Fig. 1. Architecture of a DAE. Features are extracted from the source data (e.g., noisy signals) and then mapped to a hidden representation using an encoder. A decoder synthesizes a reconstruction, where the parameters of the DAE are estimated to minimize a loss function between the DAE-synthesized data and the target data (e.g., clean signals).

structure normalization (TSN) (Ueda et al., 2016) to address limitations in the DAE input window size, global variance equalization (GVE) (Du et al., 2014) to alleviate the over-smoothing problem, and ensemble modeling (Lu et al., 2014) to learn the global and local transform functions. Recurrent neural networks (RNNs) can also be used to provide a flexible amount of temporal context while mapping the corrupted speech features to the clean speech features (Chen et al., 2015; Maas et al., 2013; Sivasankaran et al., 2015; Weninger et al., 2014). In joint training frameworks, a larger neural network structure can be built by concatenating a DNN-based front-end with a DNN-based model so that the weights in each module can be jointly adjusted to guide discrimination (Wang and Wang, 2016). For instance, DNNs can be jointly trained as a separation front-end and as a back-end model to allow error backpropagation to flow into the feature mapping layers (Narayanan and Wang, 2015). These joint training frameworks can either be sequentially trained with a specific training order (Gao et al., 2015) or simultaneously refined via multi-task learning procedures (Qian et al., 2015).

Fig. 1 shows a DAE model for data synthesis. In this section, we denote the original {source; target} features of a training set as $\{\boldsymbol{S}^X; \boldsymbol{T}^X\}$ and we denote the DAE-synthesized features as $\widehat{\boldsymbol{T}^X}$. When training a DAE model with a feature vector such as MFCC or FBANK, the source data $\boldsymbol{S}^X$ is represented by $\{\boldsymbol{s}^X(1)\ldots\boldsymbol{s}^X(l),\ldots\boldsymbol{s}^X(L)\}$ and the target data $\boldsymbol{T}^X$ is represented by $\{\boldsymbol{t}^X(1)\ldots\boldsymbol{t}^X(l),\ldots\boldsymbol{t}^X(L)\}$, where $\boldsymbol{s}^X(l)$ and $\boldsymbol{t}^X(l)$ denote the lth frame, and L denotes the total number of frames. To incorporate context information, the input vector can also be designed as $\boldsymbol{S}^X(l)=[\boldsymbol{s}^{X'}(l-\tau),\ \ldots\boldsymbol{s}^{X'}(l),\ldots\boldsymbol{s}^{X'}(l+\tau)]'$ with window length τ, while keeping the output vector as $T^X(l)=[t^X(l)]$. For a DAE with J hidden layers,

$$\begin{aligned} h^1(S^X(l)) &= sgm(W^1S^X(l)+b^1)\\ &\vdots\\ h^J\left(S^X(l)\right) &= sgm\left(W^{(J-1)}h^{(J-1)}\left(S^X(l)\right)+b^{(J-1)}\right),\\ \widehat{T^X}(l) &= W^Jh^J\left(S^X(l)\right)+b^J, \end{aligned} \tag{1}$$

where $\widehat{\boldsymbol{T}^X}(l)$ denotes the DAE-synthesized data vector, $\{\boldsymbol{W}^1\ldots\boldsymbol{W}^J\}$ are the matrices of the connection weights, and $\{\boldsymbol{b}^1\ldots\boldsymbol{b}^J\}$ are the bias vectors. The nonlinear function $sgm(.)$ of a hidden neuron is a logistic function defined as

$$sgm(t)=1/(1+exp(-t)). \tag{2}$$

The loss function is defined as

$$L\left(\widehat{T^X},\ T^X\right)=\frac{1}{L}\sum_{L}^{L}\parallel\widehat{T^X}(l)-T^X(l)\parallel_2^2 \tag{3}$$

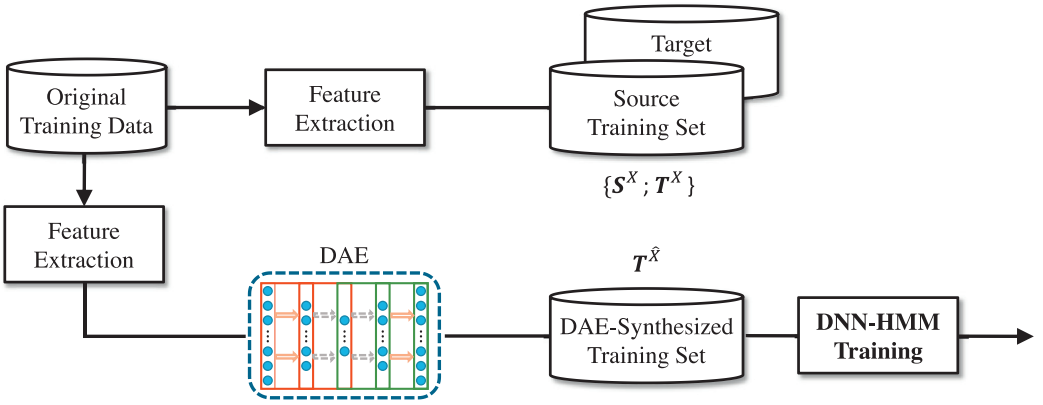

Fig. 2. Generation of a DAE-synthesized training set. These synthesized features can be directly used as the input to train a back-end DNN. This front-end procedure can also be used to derive a DAE-synthesized evaluation test set.

and the parameters of the DAE are determined by

$$\Lambda^* = \arg\min_{\Lambda}\left(L\left(T^{\widehat{X}},\ T^X\right) + \eta^1 \parallel W^1{}_{\mathrm{F}}^2 + \ldots + \eta^J W^J \parallel_{\mathrm{F}}^2\right), \tag{4}$$

where $\theta = \{\boldsymbol{W}^1 \ldots \boldsymbol{W}^J; \boldsymbol{b}^1 \ldots \boldsymbol{b}^J\}$ is the parameter set of the DAE model, $\{\eta^1 \ldots \eta^J\}$ controls the tradeoff between the reconstruction accuracy and regularization of the weighting coefficients, and $\|\cdot\|_{\mathrm{F}}^2$ denotes the Frobenius norm.

When the training process is completed, an additional training set $\widehat{\boldsymbol{T}^X}$ can be synthesized by inputting the original source data $\boldsymbol{S}^X$ into the DAE model, as shown in Fig. 2.

3. Multi-style learning

The multi-style learning procedure for a data-mixed DNN model is shown in Fig. 3. The benefits of multi-style training (Lippmann et al., 1987) can be succinctly explained in the following way in DNNs: Although lower layers implicitly seek discriminative features that are invariant across all acoustic conditions (Seltzer et al., 2013), the flexibility of sharing large amounts of parameters amongst the feature dimensions leads to over-fitting problems

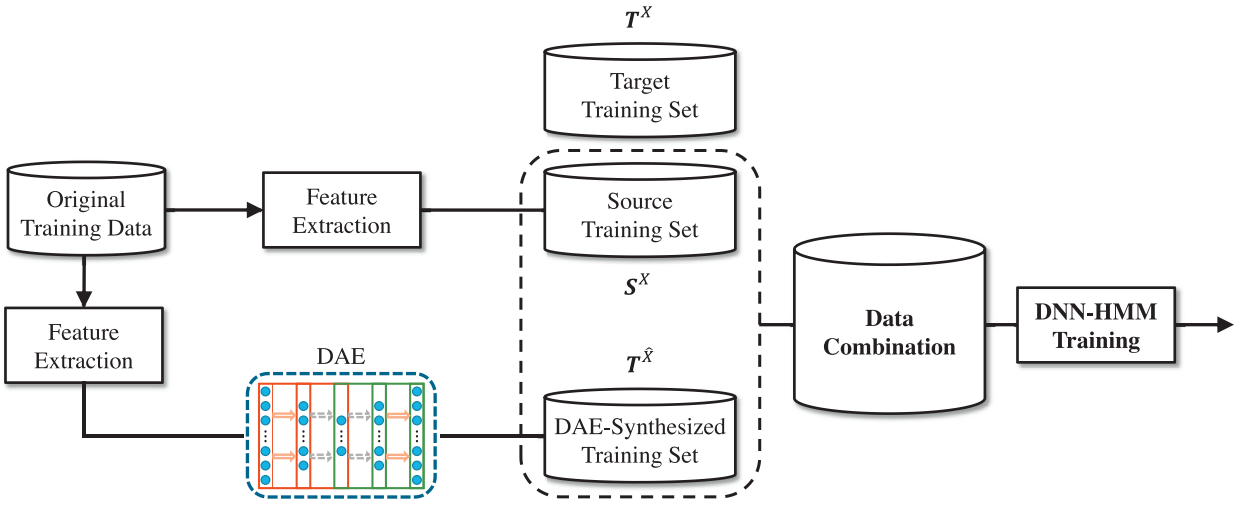

Fig. 3. Multi-style learning (Multi-style training + Deep Learning) procedure to train a data-mixed DNN. After the data combination stage, DAE-synthesized data is combined and mixed within the original source training set or within the original target training set.

in the DNN structure when training data is limited (Qi et al., 2013). Therefore, normalization techniques are usually required when the noisy target characteristics are very different from each other (Fujita et al., 2015). Injecting noises into the training data can also improve the performance on other noises (Yin et al., 2015).

Rather than injecting artificially-generated noises, the multi-style learning procedure instead calls for deep learning to be involved in generating a DAE-synthesized training set (which is later combined via DNN multi-style training). In this preliminary study, DAE-synthesized training sets were simply augmented to the original training sets. In future studies, this data combination stage can be optimized with a mixing ratio or different weights.

4. Experiments

The multi-style learning procedure will be evaluated on several tasks that are critical for unifying IoT systems: noise robustness, bandwidth extension, channel mismatch. This section covers the ASR setups.

4.1. ASR setup for the Aurora-4 noise robustness task

To evaluate noise robustness, we performed experiments on Aurora-4 (Parihar and Picone, 2002), which contains speech data from the Wall Street Journal (WSJ0) database (Paul and Baker, 1992). The 16 kHz sampling rate training set and test set were used. DNNs were all trained on Training Set 2, consisting of 7137 utterances from 83 speakers. One half of the utterances was recorded by a primary microphone, and the other half was recorded using one of a number of different secondary microphones. Both halves included clean speech and speech that was corrupted by six different types of noises (street traffic, train station, car, babble, restaurant, or airport) at 10−20 dB SNR levels. DNNs were evaluated on Test Sets 1−14, consisting of 330 utterances from 8 speakers. The test sets were recorded by either a primary microphone or by secondary microphones, and were corrupted by the six noise types used in the training set at 5−15 dB SNR. The 14 test sets are grouped into 4 subsets: (A) clean (Test Set 1), (B) noisy (Test Sets 2-7), (C) clean with channel distortion (Test Set 8), (D) noise with channel distortion (Test Sets 9-14).

This study used the Gaussian mixture model (GMM) hidden Markov model (HMM) and DNN hybrid (Hinton et al., 2012; Mohamed et al., 2012). GMM-HMMs consisted of context dependent (CD)-HMMs with 2445 senones and 16 Gaussians per state trained based on maximum likelihood (ML). The features were 13-dimensional MFCCs and cepstral mean normalization (CMN) was applied. A context window of 7 frames was concatenated to form a 91 dimensional feature vector, and linear discriminant analysis (LDA) (Fukunaga, 1972) was applied to the concatenated vector to de-correlate and perform dimensionality reduction to 40. The features were further de-correlated with maximum likelihood linear transform (MLLT) (Gopinath, 1998).

For DNNs, CMN was applied on 13 MFCCs, and these normalized features were concatenated with their first and second derivatives to form a 39 dimensional vector. A context window of 11 frames $(5 + 1 + 5)$ was used for a 39×11 visible unit input. DNN models had 6 hidden layers consisting of 2048 neurons. The output was the 2445 senones. DNNs were initialized by layer-by-layer pretraining and discriminatively trained via 18 iterations of backpropagation (using stochastic gradient descent (SGD) in mini batches of 256 examples). A learning rate of 0.008 was used for the first 7 epochs and then divided in half for each of the remaining 11 epochs. Word error rates (WER) are reported.

4.2. ASR setup for the bandwidth extension task

To evaluate bandwidth extension, we performed a series of experiments on a Mandarin Chinese broadcast news (MATBN) corpus (Wang et al., 2005). The training set (34172 utterances) and the test set (500 utterances) contained 16 kHz data with background speech, various noise types, mispronunciations, repairs, repetitions, and particles. Since MATBN is a tonal Mandarin Chinese corpus, we used a 3-state HMM for context initial and tonal final (Chen et al., 2000; Lyu et al., 2004), and a 5-state HMM for silence. Initial refers to the initial consonant of a Mandarin Chinese syllable and final refers to the vowel or diphthong part of the syllable plus an optional medial or nasal ending.

GMM-HMMs consisted of CD-HMMs with 2609 senones trained with ML estimation. 13 MFCC + 3 Pitch features were used with CMN. MFCCs were spliced in time taking a context window of 9 frames followed by

de-correlation and dimensionality reduction to 40 using LDA. The resulting features were further de-correlated with MLLT.

For DNNs, CMN was applied on 13 MFCC + 3 Pitch, and these normalized features were concatenated with their first and second derivatives to form a 48 dimensional vector. A context window of 11 frames $(5 + 1 + 5)$ was used for a 48×11 visible unit input. DNN models had 4 hidden layers consisting of 2048 neurons. The output was the 2609 senones. DNNs were initialized by layer-by-layer generative pretraining and discriminatively trained via 18 iterations of backpropagation (using SGD in mini batches of 256 examples). A learning rate of 0.008 was used for the first 7 epochs and then divided in half for each of the remaining 11 epochs. A trigram language model (Ma and Huang, 2006) was used. Character error rates (CER) are reported.

4.3. ASR setup for the channel mismatch task

To evaluate channel mismatch, we prepared a training set by combining the MATBN training set (Section 4.2) with Mtrain (a Mandarin training corpus with phonetic and tonal syllable balance, collected by ASUS). The 16 kHz Mtrain training set (2500 utterances) was recorded by 50 speakers (35 male and 15 female, 50 utterances for each speaker). Each training utterance was recorded by 7 different IoT device microphones (17,500 total utterances) denoted as Q01, Q02, Q03, Q04, Q05, Q06, Q07. The 16 kHz Mtest evaluation test set (500 utterances) was recorded by 10 speakers (5 male and 5 female, 50 utterances for each speaker). Each test utterance was also recorded by the 7 IoT microphones (3500 total utterances). Since Mtrain is a tonal Mandarin Chinese corpus, we used a 3-state HMM for context initial and tonal final, and a 5-state HMM for silence.

GMM-HMMs consisted of CD-HMMs with 4825 senones. 13 MFCC + 3 Pitch features were used with CMN. MFCCs were spliced in time taking a context window of 9 frames followed by de-correlation and dimensionality reduction to 40 using LDA. The resulting features were further de-correlated with MLLT.

For DNNs, CMN was applied on 13 MFCC + 3 Pitch, and these normalized features were concatenated with their first and second derivatives to form a 48 dimensional vector. A context window of 11 frames $(5 + 1 + 5)$ was used for a 48×11 visible unit input. DNN models had 4 hidden layers consisting of 2048 neurons. The output was the 4825 senones. DNNs were initialized by layer-by-layer generative pretraining and discriminatively trained via 18 iterations of backpropagation (using SGD in mini batches of 256 examples). A learning rate of 0.008 was used for the first 7 epochs and then divided in half for each of the remaining 11 epochs. A trigram language model (Ma and Huang, 2006) was used. Character error rates (CER) are reported.

4.4. Setup for the DAE model

The DAE structure consisted of 6 hidden layers with 2048 hidden units in each layer. The DAE output for the noise robustness, bandwidth extension, and channel mismatch tasks had 13, 16, and 16 dimensions.

For the Aurora-4 noise robustness task, the source data was Training Set 2, and the target data was Training Set 1 (clean data). The DAE input was formed from 13 MFCCs using a 13×11 context window.

For the bandwidth extension task, the source data was narrowband 8 kHz data obtained by down-sampling the wideband 16 kHz data as in Yu et al. (2013), and the target data was the original 16 kHz speech. The DAE input was formed from 13 MFCC + 3 Pitch using a 16×11 context window.

For the channel mismatch task, the source data was from the lower-quality microphones, and the target data was from Q01. The DAE input was formed from 13 MFCC + 3 Pitch using a 16×11 context window.

5. Computational results

5.1. Results for the Aurora-4 noise robustness task

In Tables 1 and 2, DAEs were setup (Section 4.4) and performance was compared for 3 training systems:

1) For **DNN—HMM baseline**, acoustic models were trained with the original training data (Training Set 2).
2) For **DAE training**, models were trained with DAE-synthesized training data (using Training Sets 1-2).
3) For **Multi-style learning**, models were trained on a mix of original and DAE-synthesized training data.

Table 1
Performance (WER%) of different training systems. The test data was the original evaluation test set of Aurora-4 for all training systems except "DAE Training and Testing" (which requires a DAE-synthesized evaluation test set). The combined average of Test Sets 1-14 is denoted as AVG.

System	A	B	C	D	AVG
DNN–HMM baseline	4.02	7.88	9.40	20.90	13.29
DAE training	3.23	22.55	18.29	41.02	28.78
DAE training and testing	3.66	7.47	8.29	19.91	12.59
Multi-style learning	3.23	6.84	9.70	19.43	**12.18**

Table 2
Clean alignment performance (WER%). The test data was the original evaluation test set of Aurora-4 for all training systems except "DAE training and testing" (which requires a DAE-synthesized evaluation test set). The combined average of Test Sets 1-14 is denoted as AVG.

System	A	B	C	D	AVG
DNN—HMM baseline	3.40	6.93	7.57	18.63	11.74
DAE training	3.12	22.31	14.40	39.22	27.62
DAE training and testing	4.00	7.31	7.88	19.51	12.34
Multi-style learning	3.14	6.14	7.32	17.15	**10.73**

Table 1 shows DAE training severely degraded the performance on the original evaluation test set. This result is expected as performance on clean speech generally degrades with noisy training. As mentioned in Yin et al. (2015), a simple approach to alleviate this problem is to involve the original speech in the DNN training. Table 1 confirms this approach, as the mix of original and DAE-synthesized training data in the proposed multi-style learning system outperformed the DAE training system. Table 1 also shows multi-style learning outperformed the DAE training and testing system, which requires front-end DAE processing on the evaluation test set. Most importantly, the proposed data-mixed DNN outperformed the DNN–HMM baseline. These results show the proposed deep learning procedure was successfully integrated into the steps of synthesizing a multi-style training set, since performance on the original test set was improved without even requiring any front-end DAE processing. The rationale for multi-style learning is similar to that of DNN noise injection (Yin et al., 2015) and is based on the ideas that DAE-synthesized patterns can be learned and thus compensated for in the inference phase, so introducing DAE-synthesized training data into the original training set can improve the generalization capability of the resulting data-mixed DNN.

In Table 2, we altered the systems to better compare with previous DNN training frameworks (e.g., Du et al., 2014; Gao et al., 2015; Qian et al., 2015) that used well-trained GMM-HMMs with features derived from Training Set 1 (clean) to perform the state-level alignment to obtain frame-level labels. Table 2 shows that even with clean GMM-HMM alignment, the multi-style learning system still outperformed the DNN–HMM baseline and both DAE training systems. Since multi-style learning only effects the synthesis of a data-mixed training set, future studies could easily integrate the procedural steps of involving original speech for either sequential (You and Xu, 2014), joint (Gao et al., 2015), or multi-task learning (Qian et al., 2015).

5.2. Results for the bandwidth extension task

In this section, the DAE regression model is specifically proposed to handle bandwidth extension. The mixed bandwidth task has practical importance as we often have access to a large amount of narrowband training data but only a small amount of wideband training data (Seltzer and Acero, 2007). In this study, narrowband training and test data was obtained by down-sampling wideband 16 kHz data to simulate 8 kHz datasets as in Yu et al. (2013). For DAEs, the source was the simulated 8 kHz speech and the target was the original 16 kHz speech.

Table 3 establishes the baselines and shows the original 16 kHz training set is severely degraded when the down-sampled 8 kHz test set is used instead of the original 16 kHz test set. A previous study (marked B1 in Li et al. (2012)) also established that DNNs perform poorly on narrowband test data when not exposed to any narrowband

Table 3
Establishing the baseline character error rate (CER%) performance on wideband (16 kHz) or narrowband (8 kHz) test sets using either wideband, narrowband, or a mix of wideband and narrowband (16 kHz + 8 kHz) training data.

Training data	CER (16 kHz)	CER (8 kHz)
16 kHz	16.18	94.93
8 kHz	–	24.97
16 kHz + 8 kHz	19.11	25.70

training data. The second baseline down-samples both the training data and test data to 8 kHz. While the performance is improved for the narrowband test set, the wideband training baseline was still optimal since wideband speech data contains additional information for distinguishing phones (Moreno and Stern, 1994). Finally, Table 3 shows performance was more uniform when DNNs were trained with a mix of both wideband and narrowband speech.

In Table 4, DAEs were setup (Section 4.4) and performance was compared for 3 training systems:

1) For **DNN–HMM baseline**, acoustic models were trained with the original 16 kHz data.
2) For **DAE training**, models were trained with the 8 kHz data that was DAE-synthesized to 16 kHz.
3) For **Multi-style learning**, models were trained on a mix of the original 16 kHz data and the 8 kHz data that was DAE-synthesized to 16 kHz.

In Table 4, performance on the 8 kHz test set that was DAE-synthesized to 16 kHz (DAE) improved the error rates (CER = 22.02) compared to the 8 kHz baseline (CER = 24.97) in Table 3, indicating that the proposed DAE bandwidth extension procedure offers immediate benefits to DNNs. In the DAE training system, performance on the 8 kHz test set that was DAE-synthesized to 16 kHz improved the error rates even further (CER = 19.02), but at the expense of degradations on the original 16 kHz test set (CER = 18.09) compared to the 16 kHz baseline (CER = 16.18) in Table 3. The AVG results in Table 4 show the proposed multi-style learning system (that exposes a mix of available 16 kHz training data and the 8 kHz data that was DAE-synthesized to 16 kHz) offers comparable performance on the original 16 kHz test set (CER = 16.47) while also maintaining the performance on the DAE-synthesized test set (CER = 19.02). Since IoT paradigms often require a unified solution for various front-ends, these multi-style learning results on both 16 kHz and DAE-synthesized speech are especially favorable.

The rationale for multi-style learning is similar to that of a recent mixed-bandwidth study (Gao et al., 2016): Back-end DNN models involve very flexible and compact structures that consist of a large amount of parameters that are highly shared among multiple feature dimensions and task targets. However, this DNN flexibility causes over-fitting problems when the training and test conditions are mismatched or when the training data is not abundant (Yin et al., 2015). Therefore in the proposed data-mixed DNNs, introducing DAE-synthesized training data into the original training set improves the generalization capability by focusing the discriminative models to simultaneously learn highly complex boundaries in heterogeneous patterns.

Table 4
Performance (CER%) on the wideband test set (16 kHz) and the synthesized (DAE) test set. The DAE test set was 8 kHz data that was DAE-synthesized to 16 kHz. The combined average of the 16 kHz test set and the (DAE) test set is denoted as AVG.

System	16 kHz	DAE	AVG
DNN–HMM baseline	16.18	22.02	19.10
DAE training	18.09	19.02	18.56
Multi-style learning	16.47	19.02	**17.75**

5.3. Results for the channel mismatch task

In Fig. 4, multi-style learning is demonstrated in a real-world IoT scenario for standardizing or incorporating a new-generation device. To build a smart world, sensor-enabled mobile devices are increasingly being connected to the internet to provide tasks and process automation (Feng et al., 2015). Since new-generation IoT devices are often added to existing ones, there is a need to adapt all of these IoT devices to support the connectivity of new services based upon the connected objects (Chaouchi, 2013). Due to the limited resources of mobile devices, data processing stages usually have to be offloaded to a central server in a mobile-sensing architecture (Higuchi et al., 2015). In Fig. 4 (right panel), the convenience of carrying out multi-style learning on the server base (offline) is demonstrated.

In this section, the DAE regression model is specifically proposed to handle channel mismatch. To prepare the data, speech samples were collected by simultaneously recording from 7 different IoT devices. Each IoT device possessed a microphone that had a specific channel characteristic or frequency response (e.g., high-quality microphones with smooth and flat frequency responses versus lower-quality microphones with nonlinear frequency responses). Fig. 5 shows the ASUS "Speech Sync" method to: (1) Synchronize the data collection from multiple devices (connected together via Bluetooth), (2) Select and align speech frames via software.

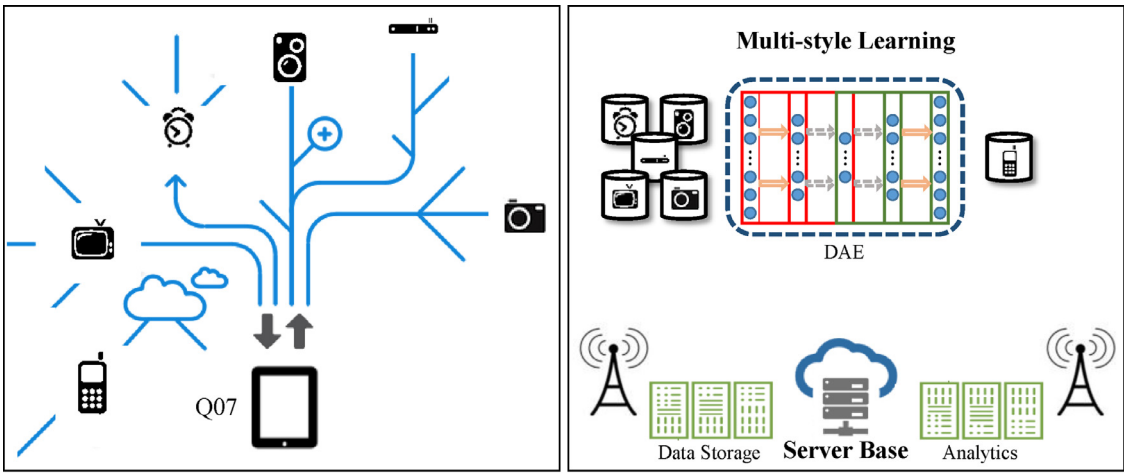

Fig. 4. Incorporating a new-generation (Q07) device (left panel). Multi-style learning on the IoT server (right panel).

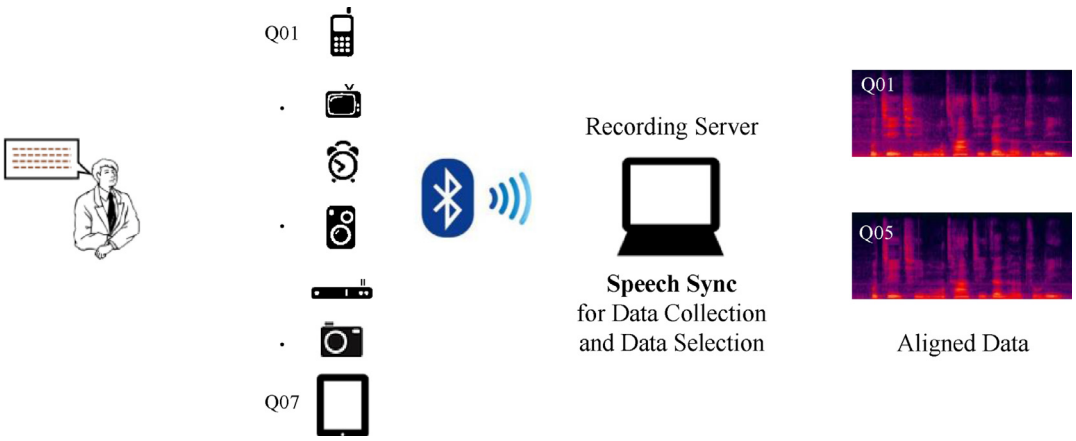

Fig. 5. Diagram of a real-world speech recording using multiple IoT devices. Each utterance was simultaneously recorded from 7 different devices (Q01, Q02,. . .Q07) at equal distances from the speaker using Speech Sync software. The example of the spectrogram from a single utterance recorded by Q01 clearly differs from that recorded by Q05.

For the channel mismatch task, the goal is to build a unified IoT system that can handle both high-quality and low-quality microphones. Table 5 establishes baseline CERs for 7 different IoT devices on the Mtest evaluation set when acoustic models were trained using the combined (MATBN and Mtrain) corpus. In Table 5, devices Q01 and Q07 achieved the best performance, and device Q05 resulted in the worst performance among the 7 IoT microphones. The average performance of devices Q01−Q07 is also listed for comparison in Table 6. For DAEs, the training data of device Q01 was selected as the target features since it produced one of the top baselines and also because previous studies conducted by ASUS indicated that this specific device was the most stable.

In Table 6 and Fig. 6, DAEs were setup (Section 4.4) and performance was compared for 3 training systems:

1) For **DNN−HMM baseline**, acoustic models were trained with the original data from devices Q01-Q07.
2) For **DAE training**, models were trained with the DAE-synthesized data that used the 6 lowest-quality device microphones (Q02, Q03, Q04, Q05, Q06, Q07) as the source features and the highest-quality device microphone (Q01) as the target features.
3) For **multi-style learning**, models were trained on a mix of the original data from Q01-Q07 and the DAE-synthesized data that used the 6 lowest-quality device microphones (Q02, Q03 Q04, Q05, Q06, Q07) as the source features and the highest-quality device microphone (Q01) as the target features.

In Table 6, average performance on the DAE-synthesized test sets (DAE) degraded (CER = 11.80) compared to the error rates on the original test sets for devices Q01−Q07 (WER = 10.84). These findings indicate the DAE front-end is not sufficient as a preprocessor without matched DAE training. In the DAE training systems, the performance on the DAE-synthesized test sets improved (CER = 10.40), but at the expense of degradations in performance on the original test sets (CER = 11.99) compared to the baselines (CER = 10.84). In the multi-style learning system, exposing a mix of available Q01−Q07 data and DAE-synthesized data improved the performance on both the original Q01−Q07 test sets (CER = 10.13) and on the DAE-synthesized test sets (CER = 9.90). Furthermore, the AVG of the multi-style learning systems (CER = 10.02) outperformed even the highest-quality Q01 microphone (CER = 10.17) in Table 5.

Table 5
Baseline performance (CER%) for devices Q01, Q02, Q03, Q04, Q05, Q06, Q07, and the combined average of Q01−Q07.

Devices	CER %
Q01	10.17
Q02	10.24
Q03	10.19
Q04	11.21
Q05	12.09
Q06	11.89
Q07	10.11
Average of Q01−Q07	10.84

Table 6
Performance (CER%) on the original test sets (Q01-Q07) and on the synthesized test sets (DAE). Results are reported as the average CER of Q01−Q07. The DAE-synthesized sets all used Q02, Q03, Q04, Q05, Q06, Q07 as the source features and Q01 as the target features. The combined average of the (Q01-Q07) test sets and the (DAE) test sets is denoted as AVG.

System	Q01−Q07	DAE	AVG
DNN—HMM baseline	10.84	11.80	11.32
DAE training	11.99	10.40	11.20
Multi-style learning	10.13	9.90	**10.02**

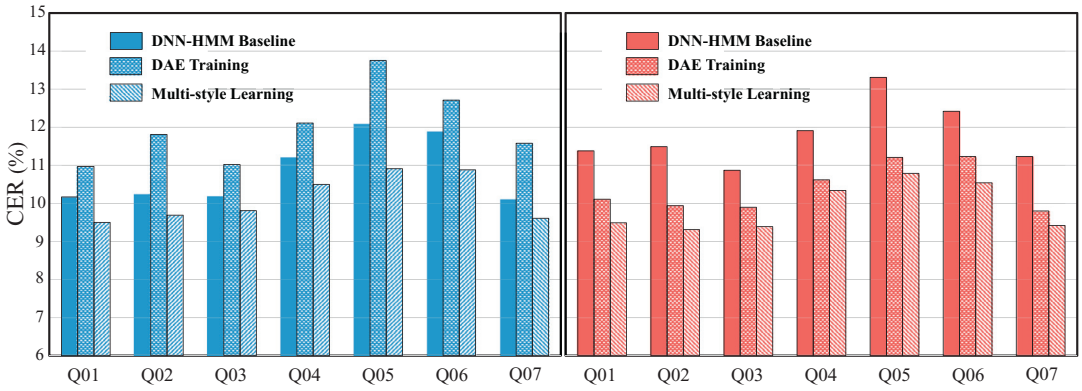

Fig. 6. Individual CERs for the 3 training procedures on each of the 7 device microphones (Q01−Q07). Each result is from systems tested on the original test data (left panel) or on the DAE-synthesized test data (right panel).

Fig. 6 presents individual results from Table 6 to compare how each training system handled each device from the original Q01−Q07 test sets and the synthesized (DAE) test sets. The left panel of Fig. 6 shows the individual results when each system was tested on the original evaluation sets, and the right panel shows the individual results for systems requiring front-end DAE processing on the evaluation sets. Although preprocessing the raw sensor data can effectively reduce the communication overhead, the computational complexity of the extracted features has to be as small as possible in IoT architectures. To address this issue, IoT industries have been pushing to design an all-in-one solution to achieve a ubiquitous networking environment. Fig. 6 shows multi-style learning provided better performance for every single device (with or without DAE preprocessing) when compared to either the DNN baseline or the DAE training frameworks. By providing a unified solution for handling many different types of devices, these results therefore highlight the efficiency of the proposed multi-style learning procedure for applications in the IoT.

Based on the preliminary results in Table 6 and Fig. 6, two additional experimental conditions were conducted. In Table 7, additional DAEs were setup (Section 4.4) and performance was compared for 3 training systems:

1) For **DNN−HMM baseline**,
   In Condition A: acoustic models were trained with the original data from devices Q01-Q06.
   In Condition B: acoustic models were trained with the original data from devices Q01 and Q03−Q07.
2) For **DAE training**,
   In Condition A: acoustic models were trained with the DAE-synthesized data that used devices Q02−Q06 as the source features and device Q01 as the target features.
   In Condition B: acoustic models were trained with the DAE-synthesized data that used devices Q03−Q07 as the source features and device Q01 as the target features.
3) For **Multi-style learning**,
   In Condition A: acoustic models were trained on a mix of the original data from devices Q01−Q06 and the DAE-synthesized data that used devices Q02−Q06 as the source features and device Q01 as the target features.
   In Condition B: acoustic models were trained on a mix of the original data from devices Q01 and Q03−Q07, and the DAE-synthesized data that used devices Q03−Q07 as the source features and device Q01 as the target features.

Table 7
Performance (CER%) on the original test set (Q07) and the synthesized test sets (DAE) for Condition A, and on the original test set (Q02) and the synthesized test sets (DAE) for Condition B. The combined average of the original test set and the (DAE) test sets is denoted as AVG.

	Condition A			Condition B		
System	Q07	DAE	AVG	Q02	DAE	AVG
DNN−HMM baseline	10.19	11.35	10.77	10.32	11.60	10.96
DAE training	11.70	9.88	10.79	11.93	10.02	10.98
Multi-style learning	9.66	9.47	**9.57**	9.75	9.37	**9.56**

In Condition A (which requires testing on a new-generation device Q07 without the availability of training data) and Condition B (which requires testing on a new-generation device Q02 without the availability of training data), the multi-style learning procedure provided better performance (with or without DAE preprocessing) when compared to either the DNN baseline or the DAE training frameworks. The results in Table 7 further highlight the effectiveness of multi-style learning for real-world scenarios in the IoT. While Table 6 demonstrated the applicability of multi-style learning for standardizing a training set, the conditions in Table 7 demonstrate the applicability of multi-style learning for incorporating a new-generation device without any available training data (as shown in Fig. 4).

In summary, the results presented throughout this section establish the utility of incorporating deep learning into traditional DNN multi-style training procedures. For IoT applications that require an efficient and unified solution for standardizing many different IoT devices or for incorporating a new-generation device, synthesizing a multi-style learning set is useful and immediately applicable. Furthermore, the proposed multi-style learning procedure addresses the aims of representation learning (Bengio et al., 2013) since it relies on multiple levels of nonlinear transformations to learn efficient mapping functions. Therefore, multi-style learning contrasts with the conventional procedures for multi-style training that only generate training data artificially (which is labor-intensive and requires human ingenuity or prior knowledge when engineering preprocessing pipelines or data transformations).

6. Conclusion

This paper analyzed a *multi-style learning* (*multi-style* training + deep *learning*) procedure for augmenting and mixing a training set for DNN-based acoustic modeling. The analysis and experiments confirmed that by combining DAE-synthesized training data with the original training data, the synthesized patterns can be effectively learned and the generalization capability of the data-mixed DNNs can be improved. Both of these advantages result in substantial performance improvements for DNN-based ASR systems that must handle noise robustness, mixed bandwidths, and channel mismatch. In this paper, we have explored three properties related to DNN−HMMs and multi-style training. First, DNNs can effectively learn multiple types of DAE-synthesized data since small perturbations in the input features are suppressed as the internal representations move to higher layers. Second, multi-style learning improves the training by combining a proportion of the original data to allow the data-mixed DNNs to focus the discriminative models to simultaneously learn highly complex boundaries in heterogeneous patterns. Third, the data-mixed DNNs can also outperform DNNs when tested on original test sets, which shows that introducing DAE-synthesized training data into the original training set improves the resulting generalization capability. All of these three properties have practical importance in IoT paradigms that require a unified solution for reducing machine error rates under all deployment environments. Furthermore, multi-style learning expands the scope of deep learning by generating additional examples of representative variations via multiple nonlinear data transformations.

Acknowledgment

Supported by Ministry of Science and Technology of Taiwan under Project MOST 105-2218-E-001-006-.

References

An, G., 1996. The effects of adding noise during backpropagation training on a generalization performance. Neural Comput. 8 (3), 643–674.
Atzori, L., Iera, A., Morabito, G., 2010. The internet of things: a survey. Comput. Netw. 54 (15), 2787–2805.
Badino, L., Canevari, C., Fadiga, L., Metta, G., 2016. Integrating articulatory data in deep neural network-based acoustic modeling. Comput. Speech Lang. 36, 173–195.
Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35 (8), 1798–1828.
Bengio, Y., Lee, H., 2015. Editorial introduction to the neural networks special issue on deep learning of representations. Neural Netw. 64, 1–3.
Bishop, C.M., 1995. Training with noise is equivalent to Tikhonov regularization. Neural Comput. 7 (1), 108–116.
Chaouchi, H., 2013. The Internet of Things: Connecting Objects. John Wiley & Sons.
Chen, B., Wang, H.M., Lee, L.-S., 2000. Retrieval of broadcast news speech in Mandarin Chinese collected in Taiwan using syllable-level statistical characteristics. In: Proceeding of ICASSP, pp. 1771–1774.
Chen, Z., Watanabe, S., Erdoğan, H., Hershey, J.R., 2015. Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks. In: Proceeding of INTERSPEECH, pp. 3274–3278.

Deng, L., Huang, X., 2004. Challenges in adopting speech recognition. Commun. ACM 47 (1), 69–75.
Deng, L., Li, X., 2013. Machine learning paradigms for speech recognition: an overview. IEEE Trans. Audio Speech Lang. Process. 21 (5), 1060–1089.
Du, J., Wang, Q., Gao, T., Xu, Y., Dai, L.-R., Lee, C.-H., 2014. Robust speech recognition with speech enhanced deep neural networks. In: Proceeding of INTERSPEECH, pp. 616–620.
Elman, J.L., Zipser, D., 1988. Learning the hidden structure of speech. J. Acoust. Soc. Am. 83 (4), 1615–1626.
Feng, X., Richardson, B., Amman, S., Glass, J., 2015. On using heterogeneous data for vehicle-based speech recognition: a DNN-based approach. In: Proceedings of ICASSP, pp. 4385–4389.
Feng, X., Zhang, Y., Glass, J., 2014. Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition. In: Proceedings of ICASSP, pp. 1759–1763.
Fujita, Y., Takashima, R., Homma, T., Ikeshita, R., Kawaguchi, Y., Sumiyoshi, T., Endo, T., Togami, M., 2015. Unified ASR system using LGM-based source separation, noise-robust feature extraction, and word hypothesis selection. In: Proceedings of ASRU, pp. 416–422.
Fukunaga, K., 1972. Introduction to Statistical Pattern Recognition. Academic Press.
Gao, T., Du, J., Dai, L.-R., Lee, C.-H., 2015. Joint training of front-end and back-end deep neural networks for robust speech recognition. In: Proceedings of ICASSP, pp. 4375–4379.
Gao, J., Du, J., Kong, K., Lu, H., Chen, E., Lee, C.-H., 2016. An experimental study on joint modeling of mixed-bandwidth data via deep neural networks for robust speech recognition. In: Proceedings of IJCNN, pp. 588–594.
Gopinath, R.A., 1998. Maximum likelihood modeling with Gaussian distributions for classification. In: Proceedings of ICASSP, pp. 661–664.
Grandvalet, Y., 2000. Anisotropic noise injection for input variables relevance determination. IEEE Trans. Neural Netw. 11 (6), 1201–1212.
Han, K., He, Y., Bagchi, D., Fosler-Lussier, E., Wang, D., 2015. Deep neural network based spectral feature mapping for robust speech recognition. In: Proceedings of INTERSPEECH, pp. 2484–2488.
Higuchi, T., Yamaguchi, H., Higashino, T., 2015. Mobile devices as an infrastructure: a survey of opportunistic sensing technology. J. Inf. Process. 23 (2), 94–104.
Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., Kingsbury, B., 2012. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. 29 (6), 82–97.
Hsiao, R., Ma, J., Hartmann, W., Karafiat, M., Grézl, F., Burget, L., Mallidi, S.H., Hermansky, H., Tsakalidis, S, Schwartz, R., 2015. Robust speech recognition in unknown reverberant and noisy conditions. In: Proceedings of ASRU, pp. 533–538.
Ishii, T., Komiyama, H., Shinozaki, T., Horiuchi, Y., Kuroiwa, S., 2013. Reverberant speech recognition based on denoising autoencoder. In: Proceedings of INTERSPEECH, pp. 3512–3516.
Kämmerer, B.R., Küpper, W.A., 1990. Experiments for isolated-word recognition with single-and two-layer perceptrons. Neural Netw. 3 (6), 693–706.
Lee, C.-H., 1998. On stochastic feature and model compensation approaches to robust speech recognition. Speech Commun. 25 (1−3), 29–47.
Leen, T.K., 1995. From data distributions to regularization in invariant learning. Neural Comput. 7 (5), 974–981.
Li, B., Sim, K.C., 2013. Improving robustness of deep neural networks via spectral masking for automatic speech recognition. In: Proceedings of ASRU, pp. 279–284.
Li, J., Yu, D., Huang, J.T., Gong, Y., 2012. Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM. In: Proceedings of IEEE SLT, pp. 131–136.
Li, J., Deng, L., Haeb-Umbach, R., Gong, Y., 2015. Robust Automatic Speech recognition: a Bridge to Practical Applications. Academic Press.
Lippmann, R.P., Martin, E.A., Paul, D.B., 1987. Multi-style training for robust isolated-word speech recognition. In: Proceedings of ICASSP, pp. 705–708.
Lu, X., Tsao, Y., Matsuda, S., Hori, C., 2013. Speech enhancement based on deep denoising autoencoder. In: Proceedings of INTERSPEECH, pp. 436–440.
Lu, X., Tsao, Y., Matsuda, S., Hori, C., 2014. Ensemble modeling of denoising autoencoder for speech spectrum restoration. In: Proceedings of INTERSPEECH, pp. 885–889.
Lyu, R.Y., Liang, M.S., Chiang, Y.C., 2004. Toward constructing A multilingual speech corpus for Taiwanese (Minnan), Hakka, and Mandarin. Int. J. Comput. Linguist. Chin. Lang. Process. 9 (2), 1–12.
Ma, N., Marxer, R., Barker, J., Brown, G.J., 2015. Exploiting synchrony spectra and deep neural networks for noise-robust automatic speech recognition. In: Proceedings of ASRU, pp. 490–495.
Ma, W.Y., Huang, C.R., 2006. Uniform and effective tagging of a heterogeneous giga-word corpus. In: Proceedings of LREC2006, pp. 24–28.
Maas, A.L, O'Neil, T.M., Hannun, A.Y., Ng, A.Y., 2013. Recurrent neural network feature enhancement: The 2nd chime challenge. In: Proceedings of Second CHiME Workshop on Machine Listening in Multisource Environments, pp. 79–80.
Matsuoaka, K., 1992. Noise injection into inputs in backpropagation learning. IEEE Trans. Syst. Man Cybern. 22, 436–446.
Mimura, M., Sakai, S., Kawahara, T., 2015. Deep autoencoders augmented with phone-class feature for reverberant speech recognition. In: Proceedings of ICASSP, pp. 4365–4369.
Mohamed, A.R., Dahl, G.E., Hinton, G., 2012. Acoustic modeling using deep belief networks. IEEE/ACM Trans. Audio Speech Lang. Process. 20 (1), 14–22.
Moreno, P.J., Stern, R.M., 1994. Sources of degradation of speech recognition in the telephone network. In: Proceedings of ICASSP, pp. 109–112.
Narayanan, A., Wang, D., 2014. Investigation of speech separation as a front-end for noise robust speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. 22 (4), 826–835.
Narayanan, A., Wang, D., 2015. Improving robustness of deep neural network acoustic models via speech separation and joint adaptive training. IEEE/ACM Trans. Audio Speech Lang. Process. 23 (1), 92–101.

Parihar, N., Picone, J., 2002. Aurora Working Group: DSR Front End LVCSR Evaluation AU/384/02. Institute for Signal and Information Process, Mississippi State University, Tech. Rep, 40, 94.
Paul, D.B., Baker, J.M., 1992. The design for the wall street journal-based CSR corpus. In: Proceedings of ICSLP, pp. 357–362.
Qi, J., Wang, D., Xu, J., Tejedor, J., 2013. Bottleneck features based on gammatone frequency cepstral coefficients. In: Proceedings of INTERSPEECH, pp. 1751–1755.
Qian, Y., Yin, M., You, Y., Yu, K., 2015. Multi-task joint-learning of deep neural networks for robust speech recognition. In: Proceedings of ASRU, pp. 310–316.
Rabiner, L., 2003. The power of speech. Science 301 (5639), 1494–1495.
Reed, R., Marks, R.J., Oh, S., 1995. Similarities of error regularization, sigmoid gain scaling, target smoothing, and training with jitter. IEEE Trans. Neural Netw. 6 (3), 529–538.
Seghouane, A.K., Moudden, Y., Fleury, G., 2002. On learning feedforward neural networks with noise injection into inputs. In: Proceedings of Neural Networks for Signal Processing, IEEE Workshop, pp. 149–158.
Seltzer, M.L., Acero, A., 2007. Training wideband acoustic models using mixed-bandwidth training data for speech recognition. IEEE Trans. Audio Speech Lang. Process 15 (1), 235–245.
Seltzer, M.L., Yu, D., Wang, Y., 2013. An investigation of deep neural networks for noise robust speech recognition. In: Proceedings of ICASSP, pp. 7398–7402.
Sietsma, J., Dow, R.J., 1991. Creating artificial neural networks that generalize. Neural Netw. 4 (1), 67–79.
Simard, P., Victorri, B., LeCun, Y., Denker, J.S., 1991. Tangent prop-a formalism for specifying selected invariances in an adaptive network. In: Proceedings of NIPS, 91, pp. 895–903.
Sivasankaran, S., Nugraha, A.A., Vincent, E., Cordovilla, J.A.M., Dalmia, S., Illina, I., Liutkus, A., 2015. Robust ASR using neural network based speech enhancement and feature simulation. In: Proceedings of ASRU, pp. 482–489.
Stankovic, J.A., 2014. Research directions for the internet of things. IEEE Internet Things J. 1 (1), 3–9.
Tsao, Y., Lu, X., Dixon, P., Hu, T.Y., Matsuda, S., Hori, C., 2014. Incorporating local information of the acoustic environments to MAP-based feature compensation and acoustic model adaptation. Comput. Speech Lang 28 (3), 709–726.
Ueda, Y., Wang, L., Kai, A., Xiao, X., Chng, E.S., Li, H., 2016. Single-channel dereverberation for distant-talking speech recognition by combining denoising autoencoder and temporal structure normalization. J. Signal Process. Syst. 82 (2), 151–161.
Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A., 2010. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. 11, 3371–3408.
Virtanen, T., Singh, R., Raj, B., 2012. Techniques for Noise Robustness in Automatic Speech Recognition. John Wiley & Sons.
Wang, H.M., Chen, B., Kuo, J.W., Cheng, S.S., 2005. MATBN: a Mandarin Chinese broadcast news corpus. Int. J. Comput. Linguist. Chin. Lang. Process. 10 (2), 219–236.
Wang, S.-S., Lin, P., Lyu, D.-C., Tsao, Y., Hwang, H.T., Su, B., 2014. Acoustic feature conversion using a polynomial based feature transferring algorithm. In: Proceedings of Chinese Spoken Language Processing (ISCSLP), pp. 454–458.
Wang, Z.Q., Wang, D., 2016. A joint training framework for robust automatic speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. 24 (4), 796–806.
Webb, A.R., 1994. Functional approximation by feed-forward networks: a least-squares approach to generalization. IEEE Trans. Neural Netw. 5 (3), 363–371.
Weng, C., Yu, D., Seltzer, M.L., Droppo, J., 2015. Deep neural networks for single-channel multi-talker speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. 23 (10), 1670–1679.
Weninger, F., Watanabe, S., Tachioka, Y., Schuller, B., 2014. Deep recurrent de-noising auto-encoder and blind de-reverberation for reverberated speech recognition. In: Proceedings of ICASSP, pp. 4623–4627.
Xie, J., Xu, L., Chen, E., 2012. Image denoising and inpainting with deep neural networks. Adv. Neural Inf. Process. Syst. 25, 341–349.
Xu, Y., Du, J., Dai, L.-R., Lee, C.-H., 2014. An experimental study on speech enhancement based on deep neural networks. IEEE Signal Process. Lett. 21 (1), 65–68.
Yin, S., Liu, C., Zhang, Z., Lin, Y., Wang, D., Tejedor, J., Zheng, T.F., Li, Y., 2015. Noisy training for deep neural networks in speech recognition. EURASIP J. Audio, Speech, Music Process. 2015 (1), 1–14.
You, Z., Xu, B., 2014. Improving wideband acoustic models using mixed-bandwidth training data via DNN adaptation. In: Proceedings of INTERSPEECH, pp. 2204–2208.
Yu, D., Seltzer, M.L., Li, J., Huang, J.T., Seide, F., 2013. Feature learning in deep neural networks-studies on speech recognition tasks. In: Proceedings of ICLR, pp. 1–9.
Yu, D., Deng, L., 2015. Feature representation learning in deep neural networks. Automatic Speech Recognition. Springer, pp. 157–175.