Training Augmentation with Adversarial Examples for Robust Speech Recognition

Sining Sun[1] , *Ching-Feng Yeh*[2], *Mari Ostendorf*[3], *Mei-Yuh Hwang*[2], *Lei Xie*[1]

[1]School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]Mobvoi AI Lab, Seattle, USA
[3]Department of Electrical Engineering, University of Washington, Seattle , USA
{snsun,lxie}@nwpu-aslp.org, {cfyeh,mhwang}@mobvoi.com, ostendor@uw.edu

Abstract

This paper explores the use of adversarial examples in training speech recognition systems to increase robustness of deep neural network acoustic models. During training, the fast gradient sign method is used to generate adversarial examples augmenting the original training data. Different from conventional data augmentation based on data transformations, the examples are dynamically generated based on current acoustic model parameters. We assess the impact of adversarial data augmentation in experiments on the Aurora-4 and CHiME-4 single-channel tasks, showing improved robustness against noise and channel variation. Further improvement is obtained when combining adversarial examples with teacher/student training, leading to a 23% relative word error rate reduction on Aurora-4.

Index Terms: robust speech recognition, adversarial examples, FGSM, data augmentation, teacher-student model

1. Introduction

In recent few years, there has been significant progress in automatic speech recognition (ASR) due to the successful application of deep neural networks (DNNs) [1, 2], such as convolutional neural networks (CNNs) [3, 4], recurrent neural networks (RNNs) [5] and sequence-to-sequence learning [5]. Acoustic modeling based on deep learning has shown robustness against noisy signals due to the deep structure and an ability to model non-linear transformations [6]. However, current ASR systems are still sensitive to environmental noise, room reverberation [7] and channel distortion. A variety of methods have been proposed to deal with noise, including front-end processing such as data augmentation [8], single or multi-channel speech enhancement [9], robust feature transformation, very deep CNNs [10, 6], and learning strategies such as teacher-student (T/S) [11] and adversarial training [12]. Here, we focus on data augmentation, assessed with and without T/S training.

The performance of acoustic models degrades when there exists a mismatch between training data and unseen test data. Data augmentation is an effective way to improve robustness, aiming to increase the diversity of the training data by augmenting it with perturbed versions using methods such as adding noises or reverberation to clean speech. Acoustic modeling with data augmentation is also known as multi-condition or multi-style training [13], which has been widely adopted in many ASR systems. Data augmentation strategies for reverberant signals was investigated in [8], which proposed using real and simulated room impulse responses to modify clean speech with different signal-to-noise ratio (SNR) levels. Generally, the robustness of the system improves with the diversity in the training data. A variational autoencoder (VAE) approach to data augmentation was proposed in [14] for unsupervised domain adaptation. They train a VAE on both clean and noisy speech without supervised information to learn a latent representation. The noisy data to be augmented are then selected by comparing the similarities with the original data, with latent representations modified by removing attributes such as speaker identity, channel and background noise.

Teacher-student (T/S) training has been widely adopted as an effective approach to increase the robustness of acoustic modeling in supervised [15] or unsupervised [11] scenarios. T/S training can also be helpful in a far-field scenario [16]. T/S training relies on parallel data to train a teacher model and a student model. For example, a close-talk data set can be used to train the teacher model, while the same speech collected by a far-field microphone can be used to train the student model. Adversarial training, which aims to learn a domain-invariant representation, is recently proposed for robust acoustic modeling [17, 18]. Adversarial training can be used to reduce the mismatches between training and test data, and it is applicable in both supervised and unsupervised scenarios.

In this paper, we propose data augmentation with adversarial examples for acoustic modeling, in order to improve robustness in adverse environments. The concept of adversarial examples was first proposed in [19] for computer vision tasks. They discovered that neural networks can easily misclassify examples in which the image pixels are only slightly skewed from the original ones. That is, the models can be very sensitive to even minor input disturbance. Adversarial examples have provoked research interest in computer vision and natural language processing [20, 21]. Recently, adversarial examples were introduced to simulate attacks to state-of-the-art end-to-end ASR systems [22]. In the work, a white-box targeted attack scenario was shown: given a natural waveform x and a nearly inaudible adversarial noise δ which is generated from x and some targeted phrase y, $x+\delta$ would be recognized as y regardless of the original content in x. This is consistent with the observation in [19] where neural networks can be vulnerable when there are minor but elaborate disturbances.

Previous work has focused on improving model robustness against adversarial test examples [20, 23]. In our work, we adopt the idea and augment training data with adversarial examples to obtain more robust acoustic models to natural data instead of adversarial examples only. In contrast to adversarial training, where the model is trained to be invariant to specific phenomena represented in the training set, the adversarial examples used here are generated automatically based on inputs

The research work is supported by the National Key Research and Development Program of China (Grant No.2017YFB1002102) and the National Natural Science Foundation of China (Grant No.61571363).

T/S training

→ Data Argumentation

回响信号

adversial attack

Adversial Examples : An input to a machine learning model
- Generated by adversial attack.
- Purposely designed to cause error
- Like a valid input for human

FGSM
Fast Gradient Sign Method

and model parameters associated with each mini-batch. In the training stage, we generate adversarial examples dynamically using the fast gradient sign method (FGSM) [20], since it has been shown to be both effective and efficient compared with other approaches for generating adversarial examples. For each mini-batch, after the adversarial examples are obtained, the parameters in the model are updated with both the original and the adversarial examples. Furthermore, we combine the proposed data augmentation scheme with teacher-student (T/S) training when parallel data is available, and find that the improvements from both approaches are additive.

The rest of the paper is organized as follows: Sec. 2 introduces adversarial examples and how to generate them using FGSM. Sec. 3 gives details of using adversarial examples for acoustic modeling. Sec. 4 describes the experimental setup and results on the CHiME-4 single track tasks and on the Aurora-4 dataset. Concluding remarks are presented in Sec. 5.

2. Adversarial examples

2.1. Definition of adversarial examples

The goal of adversarial examples is to disturb well-trained machine learning models. Relevant work shows that state-of-the-art models can be vulnerable to adversarial examples; i.e., the predictions of the models are easily misled by non-random perturbation on input signals, even though the perturbation is hardly perceptible by humans [19]. In such cases, these perturbed input signals are carefully designed and named "adversarial examples." The success of using adversarial examples to disturb models also indicates that the output distribution of neural networks may not be smooth with respect to the instances of input data distribution. As a result, a small skew in the input signals may cause abrupt changes on the output values of the models [24].

干扰

In general, a machine learning model, such as a neural network, is a parameterized function, $f(\boldsymbol{x};\boldsymbol{\theta})$, where $\boldsymbol{x}$ is the input and $\boldsymbol{\theta}$ represents the model's parameters. A trained model $f(\boldsymbol{x};\boldsymbol{\theta})$ is used to predict the label y_i given the input $\boldsymbol{x}_i$. An adversarial example $\boldsymbol{x}_i^{adv}$ can be constructed as:

$$\boldsymbol{x}_i^{adv} = \boldsymbol{x}_i + \boldsymbol{\delta}_i \tag{1}$$

so that

$$y_i \neq f(\boldsymbol{x}_i^{adv};\boldsymbol{\theta}) \tag{2}$$

where

$$\|\boldsymbol{\delta}_i\| \ll \|\boldsymbol{x}_i\|, \tag{3}$$

and $\boldsymbol{\delta}$ is called the adversarial perturbation. For a trained and robust model, small random perturbations should not have a significant impact on the output of the model. Therefore, generating adversarial perturbations as negative training examples can potentially improve model robustness.

2.2. Generating adversarial examples

In [20], the fast gradient sign method (FGSM) was proposed to generate adversarial examples using current model parameters and existing training data to generate adversarial perturbations $\boldsymbol{\delta}_i$ in equation 1.

Given model parameters $\boldsymbol{\theta}$, inputs $\boldsymbol{x}$ and the targets y associated with $\boldsymbol{x}$, the model is trained to minimize the loss function $J(\boldsymbol{\theta},\boldsymbol{x},y)$. In this work, we use average cross-entropy for $J(\boldsymbol{\theta},\boldsymbol{x},y)$, which is very common for classification tasks. Conventionally, to train a neural network, gradients are computed with the predictions of the model and the designated label, and the gradients are propagated through the layers using the back-propagation algorithm until the input layer of the network is reached. However, it is possible to further compute the gradient with respect to the input to the network (to find adversarial examples) rather than just the network weights [20].

The idea of FGSM is to generate adversarial examples that maximize the loss function $J(\boldsymbol{\theta},\boldsymbol{x},y)$,

sign of gradient matters

$$\boldsymbol{x}^{adv} = \arg\max_{\boldsymbol{x}} J(\boldsymbol{\theta},\boldsymbol{x},y) \tag{4}$$

梯度的正负影响 perturbation

$$= \boldsymbol{x} + \boldsymbol{\delta}_{FGSM}, \tag{5}$$

where

perturbation of FGSM

$$\boldsymbol{\delta}_{FGSM} = \epsilon\,\mathrm{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta},\boldsymbol{x},y)) \tag{6}$$

gradient of loss function

and ϵ is a small constant to be tuned. Note that FGSM uses the sign of the gradient instead of the value, making it easier to satisfy the constraint of equation 3. Our experimental results show that small ϵ is stable to generate adversarial examples to perturb the neural networks.

smaller ε
make the perturbation smaller

3. Training with adversarial examples

optimization:
相较传统的 data argumentation，因为 adversarial examples 被设计来让模型失误，故用其当作 X 来训练模型可能会提升 accuracy

Different from other data augmentation approaches such as adding artificial noises to simulate perturbations, the adversarial examples are generated by the model but shifted to a bigger loss value $\nabla_{\boldsymbol{x}^{adv}} J(\boldsymbol{\theta},\boldsymbol{x}^{adv},y)$ than the original data $\boldsymbol{x}$. After $\boldsymbol{x}^{adv}$ is generated, we further use it to update the model parameters, in order to enhance the robustness of the ASR system against noisy environments. In this work, FGSM is used to generate adversarial examples dynamically within each mini-batch, and the model parameters are updated immediately following the original mini-batch, as elaborated in Algorithm 1.

mini-batch:
SGD 与 BGD 的折中手段

Algorithm 1 Training neural network with automatically generated adversarial examples

Input: $D = \{\boldsymbol{x}_i, y_i\}_{k=1}^{K}$, training set
$\boldsymbol{x}_i$, input features
y_i, output labels
μ, learning rate
ϵ, adversarial weight
Output: $\boldsymbol{\theta}$, model parameters

1: Initialize model parameters $\boldsymbol{\theta}$
2: **while** model does not converge **do**
3: Read a mini-batch $B = \{\boldsymbol{x}_m, y_m\}_{m=1}^{M}$ from D
4: Train model using B,
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \frac{\mu}{M}\sum_{m=1}^{M} \nabla_{\boldsymbol{\theta}} J(\boldsymbol{x}_m, y_m, \boldsymbol{\theta})$ Update model params
5: Calculate $\{\boldsymbol{\delta}_m^{FGSM}\}_{m=1}^{M}$ using equation 6 for B
$\boldsymbol{\delta}_m^{FGSM} = \epsilon\,\mathrm{sign}(\nabla_{\boldsymbol{x}_m} J(\boldsymbol{x}_m, y_m, \boldsymbol{\theta}))$ Calculate δ_m^{FGSM}
6: Generate adversarial examples using equation 1
$\boldsymbol{x}_m^{adv} = \boldsymbol{x}_m + \boldsymbol{\delta}_m^{FGSM}$ Generate adversarial Examples
7: Make a mini-batch B_{adv} with $\{\boldsymbol{x}_m^{adv}\}_{m=1}^{M}$
$B_{adv} = \{\boldsymbol{x}_m^{adv}, y_m\}_{m=1}^{M}$
8: Train model using B_{adv},
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \frac{\mu}{M}\sum_{m=1}^{M} \nabla_{\boldsymbol{\theta}} J(\boldsymbol{x}_m^{adv}, y_m, \boldsymbol{\theta})$ Update model params by adversarial Examples
9: **end while**

In the case of acoustic modeling, the input $\boldsymbol{x}$ refers to acoustic features such as Mel-frequency cepstral coefficients (MFCCs), the label y refers to frame-level alignments such as indices of senones from forced alignments. With data augmentation using adversarial examples, we apply several steps with each mini-batch to update the model parameters: (1) Train the

model with the original inputs and obtain the adversarial perturbations $\boldsymbol{\delta}_m$ per sample as in equation 6. As ϵ is a constant, the perturbation for each feature dimension of each sample is either $+\epsilon$ or $-\epsilon$. (2) Generate adversarial examples with the obtained adversarial perturbations. (3) Make a mini-batch with adversarial examples and original labels. (4) Train the model with the mini-batch from the adversarial examples.

By feeding the original labels in the mini-batch with adversarial examples, we are minimizing the distance between the ground truth and output of the network as regular training. But what is worth noticing is that the adversarial examples are generated by FGSM to maximize the loss function as described in equation 4, which reflects the "blind spots" of the current model in input space to some extent.

adversarial examples → 模型的盲区

4. Experiments

4.1. Speech corpora and system description

4.1.1. Aurora-4 corpus

The Aurora-4 corpus is designed to evaluate the robustness of ASR systems on a medium vocabulary continuous speech recognition task based on Wall Street Journal (WSJ0, LDC93S6A). There are 7138 clean utterances from 83 speakers in the training set, recorded using the primary microphone, denoted as WSJ0 corpus here. The multi-condition training set (denoted as WSJ0m) also consists of 7138 utterances, but with a combination of clean and noisy speech perturbed by one of six different noises at 10-20 dB SNR. Channel distortion is introduced by recording the data with two different microphones. The test data set consists of four subsets: clean, noisy, clean with channel distortion, and noisy with channel distortion. The noise is simulated. The four test sets are referred to as A, B, C, D respectively in the literature. There are 330 utterances in test set A and C, and 1980 (330×6) utterances in B and D respectively. For model tuning, we use a 330-utterance subset (dev_0330) [25] of the official 1206 utterance dev set as our development set, for reasons that will be explained later. More details about Aurora-4 corpus can be found in [26].

4.1.2. CHiME-4 corpus

The CHiME-4 task is a speech recognition challenge for single-microphone or multi-microphone tablet device recordings in everyday scenarios under noisy environments. For the CHiME-4 data set, there are four noisy recording environments: street (STR), pedestrian area (PED), cafe (CAF) and bus (BUS). For training, 1600 utterances were recorded in the four noisy environments from four speakers, and additional 7138 noisy utterances simulated from WSJ0 by additive noises from the four noisy environments. The development set consists of 410 utterances in each of the four environments with both real (dt05_real) and simulated environments (dt05_simu), for a total of 3280 utterances. There are 2640 utterances in the evaluation set, with 330 utterances in each of the same eight conditions.

4.1.3. System description

We adopt convolutional neural networks (CNNs) for acoustic modeling for all the experiments we report in this work. The configurations for our CNNs are consistent with the previous work in [10], which has two convolutional layers with 256 feature maps in each layer. 9×9 filters with 1×3 pooling is used in the first layer and 3×4 filters in the second layer without pooling. There are four fully-connected layers with 1024 hidden units after the convolutional layers. Rectified linear unit (ReLU) activation function is used for all layers. For the Aurora-4 setup, 40-dimensional mel-filter bank (fbank) features with 11-frame context window are used as inputs. For the CHiME-4 setup, 40-dimensional fMLLR features with 11-frame context window are used. Standard recipes in Kaldi [27] are adopted for feature extraction, HMM-GMM training and alignment generation. As for acoustic modeling, TensorFlow [28] is used for training CNNs in this work with cross-entropy as the objective function and Adam [29] as the optimizer. The sizes for output layers are 2025 and 1942 for the Aurora-4 and CHiME-4 tasks, respectively.

As Aurora dev_0330 contains verbalized punctuations, we use the 20k closed-vocab bigram LM with verbalized punctuations during Aurora development, to fine-tune our hyper-parameter ϵ and LM weight. This dev subset is chosen so that all words are covered by the 20k vocabulary, to avoid confounding effects of noise with out-of-vocabulary issues. For the evaluation sets, we use the official WSJ 5k closed vocabulary with a 3-gram model with non-verbalized punctuations (5c-nvp 3gram), since the 5k vocabulary covers all words in the test sets.

The values of ϵ and the language weight for both tasks are tuned on the respective development sets, and the best hyper-parameters are then applied to the evaluation sets.

4.2. Experimental results

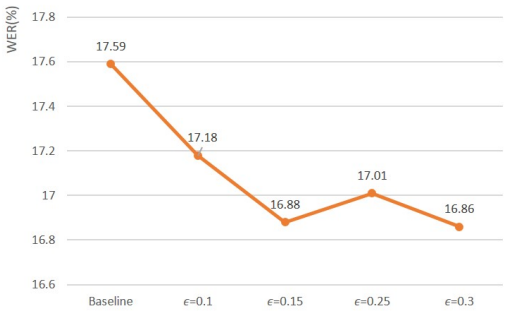

Figure 1: *WER on Aurora-4 dev_0330 with adversarial data augmentation using various perturbation weights.*

Table 1: *WER comparison on the Aurora-4 evaluation set with adversarial examples (AdvEx) ($\epsilon = 0.3$)*

	A	B	C	D	AVG.
Baseline	3.21	6.08	6.41	18.11	11.05
AdvEx	3.51	5.84	5.79	14.75	9.49
WER reduction (%)	-9.4	3.9	9.7	18.6	14.1

4.2.1. Aurora-4 results

Figure 1 shows the word error rate (WER) results on dev_0330 for different ϵ. Based on the results, $\epsilon = 0.3$ is chosen as the best perturbation weight to train the Aurora-4 model. Table 1 shows the results on the Aurora-4 evaluation set. The model trained on WSJ0m serves as the baseline. With $\epsilon = 0.3$, the augmented data training achieves 9.49% WER averaged across the four test sets, a 14.1% relative improvement over the baseline. For the test set with the highest WER on the baseline system, D, in which both noise and channel distortion are present,

Table 2: *WER comparison on CHiME-4 single-channel track evaluation sets with adversarial examples (AdvEx) ($\epsilon = 0.1$).*

system	et05_simu					et05_real				
	BUS	CAF	PED	STR	**AVE.**	BUS	CAF	PED	STR	**AVE.**
Baseline	20.25	30.69	26.62	28.74	26.57	43.95	33.64	25.95	18.68	30.55
AdvEx	19.65	29.29	24.75	26.95	**25.16**	41.00	31.34	24.74	18.23	**28.82**

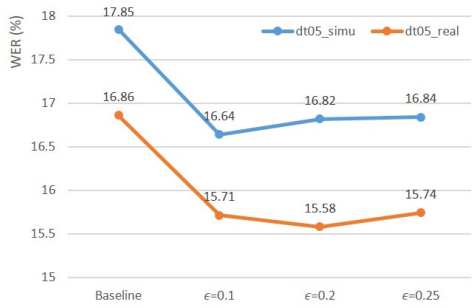

Figure 2: *WER with adversarial data augmentation on CHiME-4 dt05_simu and dt05_real data sets, with various perturbation weights.*

the proposed method reduced the WER by 18.6% relative. We also experimented with dropout training, and it gave only very small gains over the baseline.

4.2.2. CHiME-4 results

In Figure 2, different values of ϵ are selected to demonstrate the impact of ϵ on dt05_simu and dt05_real sets on CHiME-4. We see that within a reasonable range ($\epsilon < 0.25$) the proposed approach brings consistent gain. In Table 2, results on CHiME-4 single track are listed, including the real (et05_real) and simulated (et05_simu) evaluation sets. Relative WER reductions obtained on et05_real and et05_simu sets were 5.7% and 5.3%, respectively. The proposed approach was able to bring consistent improvements for all types of noises, whether in simulated or real environments.

4.2.3. Combining T/S training with data augmentation

Teacher-student (T/S) training has proven to be effective to improve the robustness of the model in scenarios where parallel data is available. As a result, in this work we also tried to combine T/S training with the proposed data augmentation method. As described in Section 4.1.1, parallel training data is available for Aurora-4. Accordingly, a teacher model is trained using clean data, while the noisy data is used to train the student model. While training the student model, the following loss function is used to optimize the parameters:

$$\begin{aligned} J_{T/S} &= \alpha\, \mathrm{CE}(y, f(\boldsymbol{x}_n, \boldsymbol{\theta}_S)) \\ &\quad + (1-\alpha)\, \mathrm{CE}(y_T, f(\boldsymbol{x}_n, \boldsymbol{\theta}_S)) \end{aligned} \tag{7}$$

where $0 < \alpha < 1$ is the discount weight, CE refers to the cross-entropy loss, y is the posterior probability estimated by the student model, $\boldsymbol{x}_n$ is a noisy (or adversarial) example, $\boldsymbol{\theta}_S$ is the student model parameter, and y_T is the posterior probability estimated from the teacher model using clean data $\boldsymbol{x}_c$,

$$y_T = f(\boldsymbol{x}_c, \boldsymbol{\theta}_T) \tag{8}$$

where $\boldsymbol{\theta}_T$ is the teacher model. The teacher model has the same configuration as the student model. This learning strategy uses a similar loss function as KullbackLeibler divergence regularization [30].

Table 3: *WER when combining T/S with adversarial data augmentation on Aurora-4.*

	A	B	C	D	AVG.
Baseline	3.21	6.08	6.41	18.11	11.05
T/S(α=0.5)	2.86	5.49	5.25	15.80	9.70
T/S+AdvEx(ϵ=0.3)	3.08	5.42	4.89	13.09	**8.50**
T/S+Random(ϵ=0.3)	3.62	5.69	5.60	14.89	9.48

Table 3 shows the WER of T/S learning with $\alpha = 0.5$. T/S learning alone gives us 12.2% relative WER reduction. After combining with adversarial data augmentation, we get the best performance with 8.50% WER.

4.2.4. Random perturbations

In order to assess the utility of data augmentation using adversarial examples, we compare the proposed approach with data augmentation using random perturbation instead of FSGM. For random perturbation, we replace $\mathrm{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ in equation 6 with a random ± 1 value. The last row of Table 3 shows that augmenting data using random perturbation gives little gain (T/S+Random) compared to using T/S learning alone. However, there is a significant performance gap between the two data augmentation methods, even though the sizes of the augmented training data are the same. This verifies the effectiveness of adversarial examples for robust acoustic modeling.

5. Conclusions

In this work, we propose data augmentation using adversarial examples for robust acoustic modeling. During training, FGSM is used to efficiently generate adversarial examples, in order to increase the diversity of the training data. Experimental results on Aurora-4 and CHiME-4 tasks show that the proposed approach can improve the robustness of acoustic modeling with deep neural networks against noise and channel variation. On the Aurora-4 evaluation set, 14.1% relative WER reduction was obtained, with the greatest benefit (18.6%) when both noise and channel distortion are present. On the CHiME-4 single track task, roughly 5% WER reductions were obtained on both real and simulated data. Similar to the use of simulated data, adversarial examples effectively increase the size of the training set without actually requiring new data. These results show that the methods are useful in combination. Adding teacher-student learning further improved performance on the Aurora-4 task, leading 23% relative WER reduction overall. Training with adversarial examples is similar in spirit to discriminative training; it would be interesting to compare and combine these approaches with different size training sets. This finding suggests that the use of adversarial examples for data augmentation is likely to be complementary to other methods for improving robustness, offering opportunities for future work.

6. References

[1] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2012.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[3] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4277–4280.

[4] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on*. IEEE, 2013, pp. 8614–8618.

[5] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[6] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2263–2276, 2016.

[7] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, A. Sehr, W. Kellermann, and R. Maas, "The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

[8] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[9] X. Xiao, C. Xu, Z. Zhang, S. Zhao, S. Sun, S. Watanabe, L. Wang, L. Xie, D. L. Jones, E. S. Chng *et al.*, "A study of learning based beamforming methods for speech recognition."

[10] S. J. Rennie, V. Goel, and S. Thomas, "Deep order statistic networks," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 124–128.

[11] J. Li, M. L. Seltzer, X. Wang, R. Zhao, and Y. Gong, "Large-scale domain adaptation via teacher-student learning," *Proc. Interspeech 2017*, pp. 2386–2390, 2017.

[12] S. Sun, B. Zhang, L. Xie, and Y. Zhang, "An unsupervised deep domain adaptation approach for robust speech recognition," *Neurocomputing*, vol. 257, pp. 79 – 87, 2017, machine Learning and Signal Processing for Big Multimedia Analysis.

[13] R. Lippmann, E. Martin, and D. Paul, "Multi-style training for robust isolated-word speech recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87.*, vol. 12. IEEE, 1987, pp. 705–708.

[14] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation," *arXiv preprint arXiv:1707.06265*, 2017.

[15] S. Watanabe, T. Hori, J. Le Roux, and J. R. Hershey, "Student-teacher network learning with enhanced features," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5275–5279.

[16] J. Yi, J. Tao, Z. Wen, and B. Liu, "Distilling knowledge using parallel data for far-field speech recognition," *arXiv preprint arXiv:1802.06941*, 2018.

[17] S. Sun, C.-F. Yeh, M.-Y. Hwang, M. Ostendorf, and L. Xie, "Domain adversarial training for accented speech recognition," *arXiv preprint arXiv:1806.02786*, 2018.

[18] Y. Shinohara, "Adversarial multi-task learning of deep neural networks for robust speech recognition." 2016.

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *Proceedings of the 2014 International Conference on Learning Representations*, Computational and Biological Learning Society, 2014.

[20] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6572

[21] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2021–2031.

[22] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," *arXiv preprint arXiv:1801.01944*, 2018.

[23] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[24] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing with virtual adversarial training," *arXiv preprint arXiv:1507.00677*, 2015.

[25] D. Pearce, "Aurora working group: Dsr front end lvcsr evaluation au/384/02," 2002.

[26] H.-G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.

[27] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[28] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 7893–7897, 2013.