Learning noise-invariant representations for robust speech recognition

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite rapid advances in speech recognition, current models remain brittle to superficial perturbations to their inputs. Small amounts of noise can destroy the performance of an otherwise state-of-the-art model. To harden models against background noise, practitioners often perform data augmentation, adding artificially-noised examples to the training set, carrying over the original label. In this paper, we hypothesize that a clean example and its superficially perturbed counterparts shouldn't merely map to the same *class* — they should map to the same *representation*. We propose invariant-representation-learning (IRL): At each training iteration, for each training example, we sample a noisy counterpart. We then apply a penalty term to coerce matched representations at each layer (above some chosen layer). Our key results, demonstrated on the LibriSpeech dataset are the following: (i) IRL significantly reduces character error rates (CER) on both 'clean' (3.3% vs 6.5%) and 'other' (11.0% vs 18.1%) test sets; (ii) on several out-of-domain noise settings (different from those seen during training), IRL's benefits are even more pronounced. Careful ablations confirm that our results are not simply due to shrinking activations at the chosen layers.

1 Introduction

Over the past several years, a series of papers have developed end-to-end deep learning systems for automatic speech recognition (ASR), advancing the state of the art on a variety of benchmarks [1, 2, 3, 4, 5, 6]. Typically, these models consist of either Recurrent Neural Networks (RNNs) with Sequence-to-Sequence (Seq2Seq) architectures [7] and attention mechanisms [8, 9], RNN transducers [10], transformer networks [11, 6], convolutional neural networks paired with transformer networks [12, 13], or RNNs trained with CTC loss [14]. Often, these models act on spectral features, e.g., Mel-Frequency Cepstral Coefficients (MFCC) [15].

While these systems achieve impressive accuracy when trained and evaluated on clean data, they suffer a well-documented sensitivity to changing noise levels and various noise types [16]. Perhaps this vulnerability should not be surprising, given the significant impact that background noise can have on MFCC features [16].

One simple strategy to combat the vulnerability of deep nets to background noise is a technique known generally as *data augmentation*, and as *multi-condition training* in the speech recognition community. Here, we augment the original data by applying transformations to which we want our models to be invariant and assigning these perturbed data points the same label as the unperturbed originals. While the computer vision literature has long focused on perturbations like random crops, rotations, translations, and Gaussian noise [17, 18, 19, 20, 21], data augmentation papers in the ASR literature commonly sample snippets of additive background noise from datasets such as MUSAN [22], which contains environmental noise (dial tones, thunder, footsteps, animal noises, etc), music

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

(baroque, classical, romantic, jazz, bluegrass, hip-hop, etc.), and speech. ASR models trained with such augmented data have demonstrated lower grapheme error rates on noisy data [23, 24, 25].

In this paper, we draw inspiration from the human ability to recognize not only that a clean clip and its noisy counterpart belong to the same category but that they are produced from the same exact recording. Thus, we propose models that map both clean inputs and their noisy counterparts onto the same point in representation space, introducing this inductive bias via regularization terms, penalizing differences between the hidden representations produced from real and noisy data. Throughout training, for each clean example, we synthesize one noisy counterpart, using a custom data augmentation pipeline that first selects a random noise snippet and volume level, adding the two raw waveforms and then generating the corresponding MFCC features on the fly. At each iteration, we apply the original cross-entropy loss on the predictions for both clean and perturbed inputs and also penalize the difference in hidden activations encouraging corresponding activations as quantified by both cosine distance and L_2 distance.

Our experiments address the LibriSpeech dataset [26], building on a Seq2Seq baseline with cross-entropy loss. To keep the empirical study clean, we do not use a language model. We run all experiments both on the standard dev and test sets and also under a variety of out-of-domain noise conditions. First, we show that while data augmentation improves generalization error on both the original task and under out-of-domain noise, the models still suffer significant degradation in performance. Next, we show that Invariant-Representation Learners (IRLs) improve significantly over generic data augmentation models, both on the *clean* and *other* (the more challenging dataset with higher word error rate) subsets of the LibriSpeech test set. Comparisons against an adversarial approach proposed by [27] and the logit pairing approach due to [28] demonstrate the significant advantage of IRL. We then demonstrate that on a variety of simulated out-of-domain noise conditions, the IRL models are considerably more robust than all baselines. Finally, we perform ablation experiments, showing that our models trained with the IRL algorithm outperform well-known regularization tactics like weight decay applied on the same representations.

1.1 Related work

A number of proposed models address the goal of noise-robust speech recognition: [29] proposes a method called *noise-aware training* that introduces information about the environment as additional inputs to DNN-based acoustic models. [23] proposes augmenting training examples with additive noise sampled from the DEMAND noise database training examples. [27] seeks noise-invariant representations in DNN-HMM architectures through an adversarial learning setup. [30] shows the training on multi-modal data leads to noise robust models. [31] demonstrates that modeling speech as a linear combination of exemplars results in noise-robust ASR models. [32] proposes deep recurrent autoencoders to denoise input features. [33] presents an overview of methods for noise-robust ASR, including recursive cepstral mean and variance normalization [34], joint adaptive training [35], and speaker adaptive training [36]. To our knowledge, no prior work in speech recognition employs our simple approach of penalizing distance between the hidden representations corresponding to clean and noisy signals.

In the most similar paper, [27] claimed that with adversarially trained DNN-HMM systems, the best performance gain is achieved when a small number of noise types are available for training. When using 6 noise classes (airport, babble, car, restaurant, street, and train), [27] found that there was no significant difference between the adversarial and baseline models. In contrast, our models show a CER improvement over baseline of 3.1% absolute on test-clean and 6.5% absolute on test-other using hundreds of noise classes.

2 Noise-invariant representations

To begin, we formally describe our loss function for enforcing noise-invariant representations on the outputs of a given layer. Because our first proposed model focuses noise-invariance in the encoding layer, we dub models using such loss functions *IRL-E*. In other experiments, we apply a cumulative penalty, additionally requiring noise-invariant representations at all subsequent layers, naming this model *IRL-C*. We begin by describing IRL-E. Subsequently, extension to IRL-C will be straightforward.

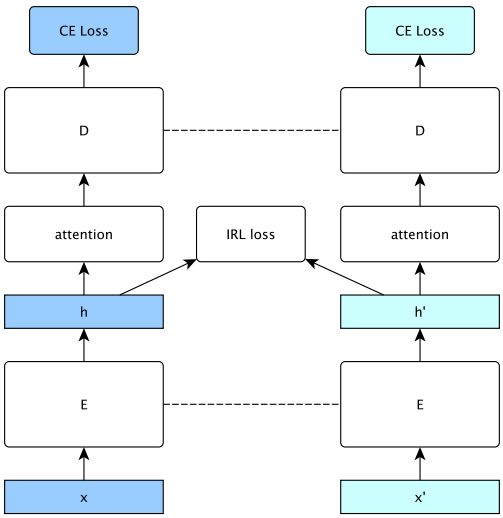


Figure 1: Diagram demonstrating the various terms in the IRL loss function as applied to a Seq2Seq attention model. Dotted lines represent shared weights.

2.1 IRL-E

The IRL algorithm is simple: First, during training, for each example $\mathbf{x}$, we produce a noisy version by sampling from $\mathbf{x}' \sim \nu(\mathbf{x})$, where ν is a stochastic function. In our experiments, this function takes a random snippet from a noise database, sets its amplitude by drawing from a normal distribution, and adds it to the original (in sample space), before converting to spectral features. We then incorporate a penalty term in our loss function to penalize the distance between the encodings of the original data point $\phi_e(\mathbf{x})$ and the noisy data point $\phi_e(\mathbf{x}')$, where ϕ_l is representation at layer l. In our experiments, we choose ϕ_e to be the output of the encoder in our Seq2Seq model. We illustrate the learning setup graphically in Figure 1. In short, our loss function consists of three terms, one to maximize the probability assigned to the the clean example's label, another to maximize the probability our model assigned to the noisy example's (identical) label scaled by hyper-parameter α, and a penalty term to induce noise-invariant representations L_d. In the following equations, we express the loss calculated on a single example $\mathbf{x}$ and its noisy counterpart $\mathbf{x}'$, omitting sums over the dataset for brevity.

$$L(\theta) = L_c(\mathbf{x}; \theta) + \alpha L_c(\mathbf{x}'; \theta) + L_d(\mathbf{x}, \mathbf{x}'; \theta),$$

where θ denotes our model parameters. Because our experiments address multiclass classification, our primary loss L_c is cross-entropy:

$$L_c(\mathbf{x}; \theta) = -\sum_{k=1}^{C} y_k \log \hat{y}_k(\mathbf{x}; \theta),$$

where C denotes the vocabulary size and $\hat{y}$ is our model's softmax output. To induce similar representations for clean and noised data, we apply a penalty consisting of two terms, the first penalizes the L_2 distance between $\phi_e(\mathbf{x})$ and $\phi_e(\mathbf{x}')$, the second penalizes their negative cosine distance.

$$L_d(\mathbf{x}, \mathbf{x}'; \theta) = \gamma \left(\phi_e(\mathbf{x}) - \phi_e(\mathbf{x}')\right)^2 - \lambda \frac{\phi_e(\mathbf{x}) \cdot \phi_e(\mathbf{x}')}{||\phi_e(\mathbf{x})|| \cdot ||\phi_e(\mathbf{x}')||}$$

We jointly penalize the L_2 and cosine distance for the following reason. It is possible to lower the L_2 distance between the two (clean and noisy) hidden representations simply by shrinking the scale of all encoded representations. Trivially, these could then be dilated again simply by setting large weights in the following layer. On the other hand, it is possible to assign high cosine similarity to the two vectors but for their magnitudes to vary significantly. By jointly penalizing L_2 and cosine distance, we require that both the clean and noisy representations point in the same direction and are close to each other in magnitude.

2.2 Applying IRL cumulatively across layers (IRL-C)

It is possible for representations to be close (but not identical) in the encoder layer, but to subsequently be pushed apart in subsequent decoder layers. Thus, we introduce another model, *IRL-C* (C for *cumulative*), that additionally applies the IRL penalty on all subsequent decoder layers. By requiring noise-invariant representations in multiple layers, we ensure that each training example and its randomly-sampled noisy counterpart have similar representations throughout the network. Note that if the encodings of the clean and noisy examples are identical at the encoder layer, then all subsequent layers will also be identical and thus those penalties will go to 0. We can express this loss as a sum over successive representations ϕ_l of the clean $\phi_l(\mathbf{x})$ and noisy $\phi_l(\mathbf{x}')$ data:

$$L_d(\mathbf{x}, \mathbf{x}'; \theta) = \sum_{l=e}^{L} \left[\gamma(\phi_l(\mathbf{x}) - \phi_l(\mathbf{x}'))^2 - \lambda \frac{\phi_l(\mathbf{x}) \cdot \phi_l(\mathbf{x}')}{||\phi_l(\mathbf{x})|| \cdot ||\phi_l(\mathbf{x}')||} \right]$$

In our experiments, we find that IRL-C consistently gives a small improvement over results achieved with IRL-E.

2.3 Application to recurrent speech models

As described to this point, our loss can be applied on any feedforward neural network with any noise process ν. Applying our technique to recurrent neural networks requires just a few additional considerations. Primarily, we must decide how to deal with the sequence structure. Two natural choices are (i) to concatenate the representations for a given layer across time steps, and then to apply our penalty on the concatenated representations and (ii) to apply the penalty separately at each time step and then to sum (or equivalently, up to a scaling factor to average) over the time steps. These approaches are identical for the L_2 penalty but not for the cosine distance penalty, owing to the normalizing factor which may be different at each time step. In this work we take approach (i) concatenating the representations across time steps and then calculating the penalty.

All of our models are based off of the sequence-to-sequence due to [9]. The input to the encoder is a sequence of spectral features, here MFCC, which are encoded by several consecutive layers of LSTM units. The encoder output states are then passed through an attention mechanism which computes the similarity between the decoder hidden states and the encoder output states. The output is a softmax over the vocabulary (here, characters) at each decoder time step.

In our experiments with IRL-E (penalty applied on a single layers), we use the output of the encoder to calculate the penalty. Note that there is one output per step in the input sequence and thus we are concatenating across the T_1 steps.

To calculate IRL-C, we also start with the encoder output concatenating across all T_1 sequence steps to calculate the IRL penalty. However, for all subsequent layers, we are acting upon layers in the decoder, and thus concatenating across the number of decoding sequence steps T_2 for calculating these terms in the IRL-C penalty.

3 Datasets

LibriSpeech We evaluate all models on the LibriSpeech [26] dataset. This dataset consists of roughly 1000 hours of audio split into training, dev and test partitions. The dataset was carefully designed to ensure that no speaker (person) appears in multiple partitions. Within both the dev and the test partitions, the data is further subdivided into "clean" and "other" subsets based on the speakers. The "clean" portion contains those speakers for which a baseline model had the lowest CER, and the "other" portion contains those speakers for whom the error rate was high. Following common practice in the literature on these datasets, we evaluate all models on the dev-clean, dev-other, test-clean, and test-other splits separately.

The MUSAN noise dataset For our additive noise, we draw upon samples from the MUSAN noise dataset [22]. MUSAN was released under a flexible Creative Commons license and consists of approximately 109 hours of noise sampled at 16kHz. The dataset contains music from several genres, namely baroque, classical, romantic jazz, bluegrass, and hip-hop, among others, speech from twelve languages, and a wide assortment of technical and non-technical noises. To generate noisy audio, we

first add MUSAN noise to the training data point at a signal-to-noise ratio drawn from a Gaussian with a mean of 12dB and a variance of 8dB. This aligns roughly with the scale of noise employed in other papers using multi-condition training [2].

4 Experiments

Before presenting our main results, we briefly describe the model architectures, training details, and the various baselines that we compare against. We also present details on our pipeline for synthesizing noisy speech and explain the experimental setup for evaluating on out-of-domain noise.

4.1 Model architecture

To facilitate reliable comparisons between our methods and various baseline training schemes, we conduct all experiments using identical architectures and tuning schemes. Because we conduct a large number of experiments and because of the computational expense of unrolling of long speech sequences, we struck a balance between performance and speed when choosing the basic architecture. The encoder for our base model consists of 4 layers: 2 encoder BLSTM layers with 320 hidden units each, followed by 2 encoder LSTM layers with 320 hidden units each. Our decoder accesses the encoded representations using dot product attention, and contains 4 decoder LSTM layers, with 320 hidden units each. Notably, our first encoder layer halves the sequence length by concatenating adjacent inputs along the temporal axis. Each model across all of our comparisons has the exact same number of trainable parameters. To keep things simple, we do not use an external language model. Instead we decode predictions from all models via beam search with width 10.

To ensure fair comparisons, we perform hyper-parameter searches separately for each model and account for variability due to initialization by training each model 5 times and keeping the best run as determined on the dev-other partition. Specifically, we tune the weights on our losses by trying each of the scale values (0.001, 0.01, 0.1, 1, 10, and 100). We found that an α of 1 (the weight on the cross-entropy loss of the noised data), a γ of 0.01 (the weight on the L2 distance loss), and a λ of 0.01 (the weight on the cosine distance loss) worked well.

4.2 Training details

We train all models with the Adam optimizer with an initial learning rate of 0.001. We employ a learning rate schedule similar to NewBob [37] that decreases by a factor of 2 if there is an increase in validation perplexity epoch-over-epoch. We employ a stopping criterion that ends training if validation perplexity does not decrease for three epochs in a row. We limit each models to a maximum of 40 epochs, although our networks generally converge within 20 epochs.

The primary loss function for each model is cross-entropy loss and our primary evaluation metric to evaluate all models is the character error rate. As described above, the additional loss terms for our IRL models are L2 loss and cosine distance between representations of clean and noisy audio.

4.3 Baselines

Our baseline models include a model trained on the standard training data, a model trained with noise-augmented data, a model trained with noise augmented data and weight decay, and a data augmented model supervised with L2 loss to push activations of the encodings to 0. These ablation tests provide evidence that our IRL algorithm isn't simply penalizing the norm of the encodings.

- **Baseline:** Our base model trains the baseline sequence-to-sequence model on the original 960 hours of LibriSpeech training data.
- **Data augmentation:** Our data augmentation model trains the sequence-to-sequence model on both the examples from the 960 hour LibriSpeech training corpus and the randomly generated noisy counterparts.
- **Adversarial:** The adversarial model consists of an adversarial noise discriminator trained on top of the encoder outputs. The discriminator consists of 2 layers of 256 ReLu units and a single unit sigmoid output. We train the discriminator to classify whether the representation originates from clean or noised inputs. The encoder meanwhile is trained both to minimize

the classification loss and to fool the discriminator, in a scheme similar to the reverse gradient technique in the domain-adversarial approach due to [38] and applied to speech by [27].

- **Logit pairing:** Our final baseline consists of the logit pairing model due to [28] which applies L2 loss and cosine distance loss on the final decoder layer logits, enforcing noise-invariant representations but only on the output layer.

4.4 Synthesizing noise

We train all models on the LibriSpeech corpus, generating noisy data by adding randomly selected noise tracks from the MUSAN dataset with a signal to noise ratio drawn from a Gaussian distribution (12dB mean, 8dB standard deviation) and temporal shift drawn from a uniform distribution (with a range of 0 to 1000ms). For the data augmentation model, this result resembles the typical data augmentation (multi-condition training) procedure.

4.5 Out-of-domain noise

Next, we evaluate each of our models on a variety of noise conditions that were not seen at training time. In particular, we consider the following out-of-domain noise conditions: (i) augmenting the test-clean split with overlapping out-of-domain speech from the WSJ-0 dataset [39] to simulate multi-speaker environments, (ii) applying additive noise with various SNRdb to simulate varying noise levels, (iii) modulating the volume of the clean signal to simulate different levels of speaker loudness, (iv) convolving the original wave file with room impulse responses to simulate the effect of room reverberation on speech, and (v) re-sampling to 8kHz to simulate telephony data. For each setting, we measure CER on the out-of-domain noise-augmented test-clean data.

5 Results

Table 1: Evaluation and test set character error rate on the LibriSpeech corpus.

	Evaluation set		**Test set**	
	dev-clean	dev-other	test-clean	test-other
Baseline	6.7%	17.8%	6.5%	18.1%
Data aug.	6.4%	16.8%	6.4%	17.5%
Adversarial	6.7%	16.7%	6.5%	17.6%
Logit pairing	5.1%	14.5%	5.1%	14.8%
IRL-E	3.6%	11.0%	3.5%	11.2%
IRL-C	**3.4%**	**10.7%**	**3.3%**	**11.0%**

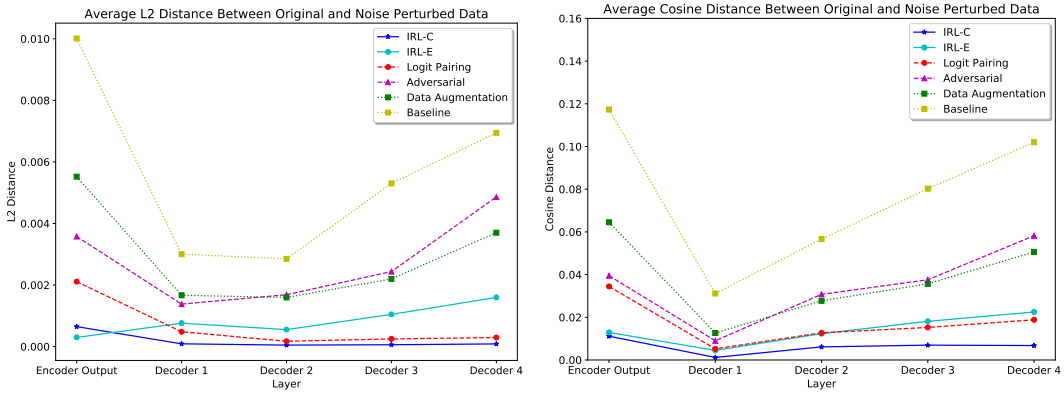
Our IRL-C model achieves the best CER on both test-clean and test-other 3.3% and 11%, respectively (Table 1). This compares baseline scores of 6.5% and 18.1%, respectively. We note that by comparison, conventional data augmentation is only marginally effective. Among the baselines that we consider, logit pairing performs best (5.1% and 14.8%) although the improvements are not comparable to either IRL model.

We found that weight decay slowed down network convergence and did not outperform pure data augmented training. However, [40] showed that weight decay is most effective with separate λ_{rec} and λ_{nonrec} hyper-parameters for determining the strength of regularization for the recurrent and non-recurrent weight matrices. We have not tried this in our experiments. Additionally, we discovered that applying multi-condition training while naively lowering the activations of hidden representations leads to nearly identical performance (on both the original and out-of-domain noise perturbed test data) and convergence trajectory as the base model trained on noise augmented data. These results support our hypothesis that models trained with the IRL algorithm do not trivially decrease the magnitude of intermediate representations.

Our final experiments test the effects of various out-of-domain noise on our models. The results are shown in Table 2. We found that our models trained with the IRL procedure had stronger results (and significantly less degradation) across all tasks compared to the baseline and the purely data

Table 2: Character error rate for test-clean augmented with noise

	CER on noisy data					
	Base	Data aug.	Adv.	Logit	IRL-E	IRL-C
Error on test-clean	6.5%	6.4%	6.5%	5.1%	3.5%	**3.3%**
In-domain (6SNRdB)	27.8%	10.8%	16.5%	8.7%	6.0%	**5.7%**
In-domain (12SNRdB)	13.5%	7.8%	12.1%	6.2%	4.2%	**4.1%**
Impulse convolve	24.1%	21.0%	28.3%	47.6%	18.0%	**13.8%**
Speech (6SNRdB)	91.5%	32.0%	67.7%	33.0%	16.4%	**14.1%**
Speech (12SNRdB)	77.8%	15.2%	34.7%	11.1%	7.6%	**6.8%**
Volume (+6 dB)	6.5%	6.4%	9.8%	5.1%	3.6%	**3.5%**
Volume (−6 dB)	6.5%	6.3%	9.6%	5.0%	3.6%	**3.5%**
Telephony	14.2%	12.2%	21.3%	10.3%	7.1%	**6.4%**



(a) L2 distance

(b) Cosine distance

Figure 2: Average distance between original and noised data for various models (distinct lines) and various layers (x-axis). Subplot (a) depicts L2 distance and (b) depicts cosine distance.

augmented models. When applying various room reverberation on speech, we found that the IRL-C model had a character error rate of 13.8% compared to 21.0% on the data augmented model and 24.1% on the baseline model. Our IRL-C model shows 14.1% character error rate on out-of-domain overlapping speech compared to 91.5% for the baseline and 32.0% on the data augmented model. We found that decreasing the signal-to-noise ratio also effected the baseline models more than the models trained on the IRL algorithm: our IRL-C model received a character error rate of 5.7% compared to 27.8% for baseline and 10.8% for the purely data augmented model. We found that modifying the volume of the speaker did not effect the accuracy of any of the networks. Finally, we found that our models trained with the IRL algorithm performed better for re-sampled telephony data, achieving a character error rate of 6.4% for IRL-C compared to 14.2% for baseline and 12.2% for the purely data augmented model.

We also executed some empirical analysis to determine the effect of the various approaches on the distances between noisy examples and their clean counterparts in representation space. In general, our IRL models have the lowest L2 and cosine distances between noisy representations and the clean counterparts. In Figure 2, you can see that although the IRL-E and IRL-C model models have similarly close representations at the encoder layer, neither reaches 0 distance. Then for IRL-E over the subsequent layers, the clean and noisy representations diverge again, while for IRL-C they remain close throughout.

6 Conclusions

In this paper, we demonstrated that enforcing noise-invariant representations by penalizing differences between pairs of clean and noisy data can increase model accuracy on the ASR task, produce models that are robust to out-of-domain noise, and improve convergence speed. The performance gains achieved by IRL come without any impact to inference throughput. We note that our core ideas here can be applied broadly to deep networks for any supervised task. While the speech setting is particularly interesting to us, our methods are equally applicable to other machine learning fields, notably computer vision. One natural extension might be to experiment with various other loss functions such as triplet losses, requiring that noisy data be both close to its clean counterpart and further away from *different* clean data. Additionally, our approach might be well-suited to conferring greater robustness to adversarial examples. The comparative improvements over requiring invariant hidden representations vs. invariant logits here raises the possibility that we might be able to realize similar gains over logit pairing in the adversarial setting.

References

[1] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., "Deep speech: Scaling up end-to-end speech recognition," p. arXiv:1412.5567, 2014.

[2] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, and et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," p. arXiv:1512.02595, 2015.

[3] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," 2015.

[4] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[5] Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney, "Improved training of end-to-end attention models for speech recognition," *Interspeech*, 2018.

[6] Shiyu Zhou, Linhao Dong, Shuang Xu, and Bo Xu, "Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese," *arXiv preprint arXiv:1804.10752*, 2018.

[7] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014.

[8] Minh-Thang Luong, Hieu Pham, and Christopher D Manning, "Effective approaches to attention-based neural machine translation," p. arXiv:1508.04025, 2015.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint, 2014, p. arXiv:1409.0473.

[10] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[12] Shiyu Zhou, Linhao Dong, Shuang Xu, and Bo Xu, "A comparison of modeling units in sequence-to-sequence speech recognition with the transformer on mandarin chinese," *arXiv preprint arXiv:1805.06239*, 2018.

[13] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve, "Wav2letter: An end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[14] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013.

[15] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," IEEE Transactions on Acoustics Speech and Signal Processing, 1980.

[16] U. Bhattacharjee, Swapnanil Gogoi, and Rubi Sharma, "A statistical analysis on the impact of noise on mfcc features for speech recognition," *2016 International Conference on Recent Advances and Innovations in Engineering*, 2016.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[18] Guozhong An, "The effects of adding noise during backpropagation training on a generalization performance," *Neural computation*, 1996.

[19] Yves Grandvalet and Stéphane Canu, "Comments on" noise injection into inputs in back propagation learning"," *IEEE Transactions on Systems, Man, and Cybernetics*, 1995.

[20] Chris M Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural Computation*, 1995.

[21] Yves Grandvalet, Stéphane Canu, and Stéphane Boucheron, "Noise injection: Theoretical prospects," *Neural Computation*, 1997.

[22] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," arXiv preprint, 2015, p. arXiv:1510.08484v1.

[23] S. Yin, C. Liu, Z. Zhang, Y. Lin, D. Wang, J. Tejedor, T. Zheng, and Y. Li, "Noisy training for deep neural networks in speech recognition," Journal on Audio, Speech, and Music Processing, 2015.

[24] Hong Yu, Achintya Sarkar, Dennis Alexander Lehmann Thomsen, Zheng-Hua Tan, Zhanyu Ma, and Jun Guo, "Effect of multi-condition training and speech enhancement methods on spoofing detection," IEEE First International Workshop on Sensing, Processing and Learning for Intelligent Machines, 2016.

[25] J Rajnoha, "Multi-condition training for unknown environment adaptation in robust asr under real conditions," *Acta Polytechnica*, 2009.

[26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," IEEE International Conference on Acoustics, Speech and Signal Processing, 2015.

[27] D. Serdyuk, K. Audhkhasi, P. Brakel, B. Ramabhadran, S. Thomas, and Y. Bengio, "Invariant representations for noisy speech recognition," Computing Research Repository, 2016, p. abs/1612.01928.

[28] H. Kannan, A. Kurakin, and I. Goodfellow, "Adversarial logit pairing," arXiv preprint, 2018, p. arXiv:1803.06373.

[29] Michael L Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013.

[30] Jing Huang and Brian Kingsbury, "Audio-visual deep learning for noise robust speech recognition," IEEE International Conference on Acoustics, Speech and Signal Processing, 2013.

[31] Jort F Gemmeke, Tuomas Virtanen, and Antti Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[32] Andrew L Maas, Quoc V Le, Tyler M O'Neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y Ng, "Recurrent neural networks for noise reduction in robust asr," in *Conference of the International Speech Communication Association*, 2012.

[33] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, 2014.

[34] Olli Viikki, David Bye, and Kari Laurila, "A recursive feature vector normalization approach for robust speech recognition in noise," IEEE International Conference on Acoustics, Speech and Signal Processing, 1998.

[35] Hank Liao and MJ F Gales, "Adaptive training with joint uncertainty decoding for robust recognition of noisy data," IEEE International Conference on Acoustics, Speech and Signal Processing, 2007.

[36] Tasos Anastasakos, John McDonough, Richard Schwartz, and John Makhoul, "A compact model for speaker-adaptive training," IEEE International Conference on Spoken Language, 1996.

[37] ICSI Berkeley, "Quicknet," *Available: http://www1.icsi.berkeley.edu/Speech/qn.html*, 2000.

[38] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, 2016.

[39] et al. Garofolo, John S., "Csr-i (wsj0) complete ldc93s6a.," *Web Download. Philadelphia: Linguistic Data Consortium*, 1993.

[40] Markus Kliegl, Siddharth Goyal, Kexin Zhao, Kavya Srinet, and Mohammad Shoeybi, "Trace norm regularization and faster inference for embedded speech recognition rnns," *arXiv preprint arXiv:1710.09026*, 2017.