Recurrent Neural Networks for Noise Reduction in Robust ASR

Andrew L. Maas[1], *Quoc V. Le*[1], *Tyler M. O'Neil*[1], *Oriol Vinyals*[2], *Patrick Nguyen*[3], *Andrew Y. Ng*[1]

[1]Computer Science Department, Stanford University, CA, USA
[2]EECS Department, University of California - Berkeley, Berkeley, CA, USA
[3]Google, Inc., Mountain View, CA, USA

[amaas,quocle,toneil]@cs.stanford.edu, vinyals@eecs.berkeley.edu,
drpng@google.com, ang@cs.stanford.edu

Abstract

Recent work on deep neural networks as acoustic models for automatic speech recognition (ASR) have demonstrated substantial performance improvements. We introduce a model which uses a deep recurrent auto encoder neural network to denoise input features for robust ASR. The model is trained on stereo (noisy and clean) audio features to predict clean features given noisy input. The model makes no assumptions about how noise affects the signal, nor the existence of distinct noise environments. Instead, the model can learn to model any type of distortion or additive noise given sufficient training data. We demonstrate the model is competitive with existing feature denoising approaches on the Aurora2 task, and outperforms a tandem approach where deep networks are used to predict phoneme posteriors directly.

Index Terms: neural networks, robust ASR, deep learning

1. Introduction

Robust automatic speech recognition (ASR), that with background noise and channel distortion, is a fundamental problem as ASR increasingly moves to mobile devices. Existing state-of-the-art methods for robust ASR use specialized domain knowledge to denoise the speech signal [1] or train a word-segment discriminative model robust to noise [2]. The performance of such systems is highly dependent upon the methods' designers imbuing domain knowledge like temporal derivative features or signal filters and transforms.

Neural network models have also been applied to learn a function to map a noisy utterance $\mathbf{x}$ to a clean version of the utterance $\mathbf{y}$ [3]. Such function approximators encode less domain-specific knowledge and by increasing hidden layer size can approximate increasingly nonlinear functions. Previous work also introduced neural network models which capture the temporal nature of speech using recurrent connections, and directly estimate word identity for isolated digit recognition [4]. However, previous work has focused on fairly small neural network models and experimented with simple noise conditions – often a single background noise environment.

Our novel approach applies large neural network models with several hidden layers and temporally recurrent connections. Such *deep* models should better capture the complex relationships between noisy and clean utterances in data with many noise environments and noise levels. In this way, our work shares underlying principles with deep neural net acoustic models, which recently yielded substantial improvements in ASR [5]. We experiment with a reasonably large set of background noise environments and demonstrate the importance of models with many hidden layers when learning a denoising function.

Training noise reduction models using stereo (clean and noisy) data has been used successfully with the SPLICE algorithm [6]. SPLICE attempts to model the joint distribution between clean and noisy data, $p(\mathbf{x}, \mathbf{y})$. Separate SPLICE models are trained for each noise condition and noise level. Thus beyond stereo training data, SPLICE also requires noise settings to be labeled or automatically clustered to train separate models for each condition. SPLICE is similar to a linear neural network with fixed non-linear basis functions. Our model attempts to learn the basis functions with multiple hidden layers as opposed to choosing nonlinearities which may be important *a priori*.

We demonstrate our model's ability to produce cleaned utterances for the Aurora2 robust ASR task. The method outperforms the SPLICE denoising algorithm, as well as the hand-engineered ETSI2 advanced front end (AFE) denoising system. We additionally perform control experiments to assess our model's ability to generalize to an unseen noise environments, and quantify the importance of deep and temporally recurrent neural network architecture elements.

2. Model

For robust ASR we would like a function $f(\mathbf{x}) \rightarrow \mathbf{y}$ which maps the noisy utterance $\mathbf{x}$ to a clean utterance $\mathbf{y}$ with background noise and distortion removed. For example we might hand engineer a function $f(\mathbf{x})$ using band pass filters to supress the signal at frequencies we suspect contain background noise. However engineering such functions is difficult and often doesn't capture the rich complexity present in noisy utterances. Our approach instead learns the function $f(\mathbf{x})$ using a broad class of nonlinear function approximators – neural networks. Such models adapt to model the nonlinear relationships between noisy and clean data present in given training data. Furthermore, we can construct various network architectures to model the temporal structure of speech, as well as enhance the nonlinear capacity of the model by building *deep* multilayer function approximators.

Given the noisy utterance $\mathbf{x}$, the neural network outputs a prediction $\hat{\mathbf{y}} = f(\mathbf{x})$ of the clean utterance $\mathbf{y}$. The error of the output is measured via squared error,

$$||\hat{\mathbf{y}} - \mathbf{y}||^2, \tag{1}$$

where $|| \cdot ||$ is the ℓ_2 norm between two vectors. The neural network learns parameters to minimize this error on a given training set. Such noise removal functions are often studied in the context of specific feature types, such as cepstra. Our model is instead agnostic as to the type of features present in $\mathbf{y}$ and can thus be applied to any sort of speech feature without modification.

2.1. Single Layer Denoising Autoencoder

A neural network which attempts to reconstruct a clean version of its own noisy input is known in the literature as a denoising autoencoder (DAE) [7]. A single hidden layer DAE outputs its prediction $\hat{y}$ using a linear reconstruction layer and signle hidden layer of the form,

$$\hat{y} = Vh^{(1)}(x) + c \tag{2}$$

$$h^{(1)}(x) = \sigma(W^{(1)}x + b^{(1)}) \tag{3}$$

The weight matrices V and $W^{(1)}$ along with the bias vectors c and $b^{(1)}$ parameters of the model. The hidden layer representation $h^{(1)}(x)$ is a nonlinear function of the input vector x because $\sigma(\dot{})$ is a point-wise nonlinearity. We use the logistic function $\sigma(z) = \frac{1}{1+e^z}$ in our work.

Because an utterance $\mathbf{x}$ is variable-length and that training an autoencoder on a high-dimensional input is expensive, it is typical to resort to train an DAE on a small temporal context windows. This increases computational efficiency and saves the model from needing to re-learn the same denoising function at each point in time. Furthermore, this technique allows the model to handle large variation in utterance durations without the need to zero pad inputs to some maximum length. Ultimately the entire clean utterance prediction $\hat{\mathbf{y}}$ is created by applying the DAE at each time sample of the input utterance – much in the same way as a convolutional filter.

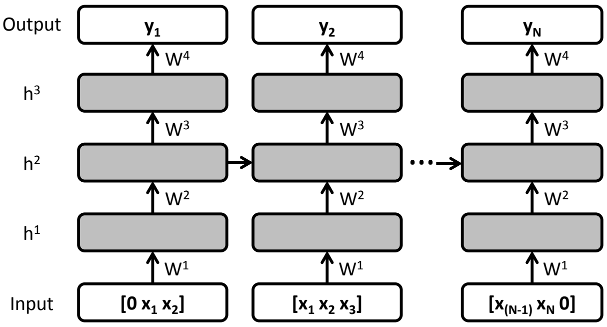

Figure 1: *Deep Recurrent Denoising Autoencoder. A model with 3 hidden layers that takes 3 frames of noisy input features and predicts a clean version of the center frame*

2.2. Recurrent Denoising Autoencoder

The conventional DAE assumes that only short context regions are needed to reconstruct a clean signal, and thus considers small temporal windows of the utterance independently. Intuitively this seems a bad assumption since speech and noise signals are highly correlated at both long and short timescales. To address this issue we add temporally recurrent connections to the model, yielding a recurrent denoising autoencoder (RDAE). A recurrent network computes hidden activations using both the input features for the current timestep x_t and the hidden representation from the previous timestep $h^{(1)}(x_{t-1})$. The full equation for the hidden activation at time t is thus,

$$h^{(1)}(x_t) = \sigma(W^{(1)}x_t + b^{(1)} + Uh^{(1)}(x_{t-1})), \tag{4}$$

which builds upon the DAE (Equation 3) by adding a weight matrix U which connects hidden units for the current timestep to hidden unit activations in the previous timestep. The RDAE thus does not assume independence of each input window x but instead models temporal dependence which we expect to exist in noisy speech utterances.

2.3. Deep Architectures

A single layer RDAE is a nonlinear function, but perhaps not a sufficiently expressive model to capture the complexities of noise environments and channel distortions. We thus make the model more nonlinear and add free parameters by adding additional hidden layes. Indeed, much of the recent success of neural network acoustic models is driven by *deep* neural networks – those with more than one hidden layer. Our models naturally extend to using multiple hidden layers, yielding the deep denoising autoencoder (DDAE) and the deep recurrent denoising autoencoder (DRDAE). Figure 1 shows a DRDAE with 3 hidden layers. Note that recurrent connections are only used in the middle hidden layer in DRDAE architectures.

With multiple hidden layers we denote the i^{th} hidden layer's activation in response to input as $h^{(i)}(x_t)$. Deep hidden layers, those with $i > 1$ compute their activation as,

$$h^{(i)}(x_t) = \sigma(W^{(i)} h^{(i-1)}(x_t) + b^{(i)}). \qquad (5)$$

Each hidden layer $h^{(i)}$ has a corresponding weight matrix $W^{(i)}$ and bias vector $b^{(i)}$. For recurrent models, the middle hidden layer has temporal connections as in Equation 4.

3. Experiments

We perform robust ASR experiments using the Aurora2 corpus [?]. Noisy utterances in Aurora2 were created synthetically, so it provides noisy and clean versions of the same utterance which is required to train our denoising models. However, the training set provides only four noise environments and we do not expect our model to learn a general denoising function given such a limited view of possible clean utterance corruptions. Our model attempts to predict clean MFCC features given MFCC features of the corrupted utterance.

3.1. Training

We train our model using the standard multi-condition training set which includes 8,001 utterances corrupted by 4 noise types at 5 different noise levels. Model gradients are computed via backpropagation – unrolling the model through time with no truncation or approximation for recurrent models. The L-BFGS quasi-Newton optimization algorithm is used to train all models from random initialization. We find this batch optimization technique to perform as well as the pre-training and stochastic gradient fine-tuning approach used in other deep learning work [8]. We train a DRDAE with 3 hidden layers of 500 hidden units and use an input context of 3 frames to create x_t.

3.2. Robust ASR

We evaluate the model using the "clean" testing condition – where the standard HMM system is trained on only clean data and evaluated on noisy data. This condition evaluates whether the model can properly transform noisy features into their corresponding clean versions as expected by the HMM acoustic models.

Table 1 shows error rates for each noise condition averaged across the 4 noise types present in test set A. For comparison we include the error rates when using the original MFCC features, as well as the features produced by the ETSI2 advanced front end (AFE) denoising system [1]. Overall, our model outperforms the MFCC baseline as well as the AFE denoising system. At lower SNRs our model reconstructs predicts clean MFCCs much better than the AFE leading to substantial performance improvements. When averaged across SNR conditions and ignoring the clean and -5dB settings as per the ETSI standard, our model gives a WER of 10.85%. This is substantially better than the 12.26% of the AFE, and outperforms the SPLICE denoising algorithm which yields an 11.67% WER for this setting.

SNR	MFCC	AFE	Tandem	DRDAE
Clean	0.94	**0.77**	0.79	0.94
20dB	5.55	1.70	2.21	**1.53**
15dB	14.98	3.08	4.57	**2.25**
10dB	36.16	6.45	7.05	**4.05**
5dB	64.25	14.16	17.88	**10.68**
0dB	84.62	35.92	48.12	**32.90**
-5dB	90.04	**68.70**	79.54	70.16

Table 1: Word error rates (WER) on Aurora2 test set A. Performance is averaged across the four noise types in the test set. These noise environments are the same four present in the DRDAE training set.

We also compare to a tandem approach, where a neural network is trained to output phoneme posteriors as opposed to our model which outputs a cleaned feature vector. In the tandem setup, as in our own, network output is used as observations for the HMM Gaussian mixutre models. We find our model to perform substantially better than a recursive neural network used in a tandem setup as reported by [?], whose results are also shown in Table 1. This suggests that for robust ASR it is better to predict clean features rather than predicting phoneme posteriors directly. It is important to note however that the tandem network was trained only on clean speech, and that a tandem approach trained on the multi-condition training set could perform better.

3.3. ASR on Unseen Noise Environments

Test set A uses the same four noise environments as the training set used to train our DRDAE model. Because the model discriminatively trains to remove only the four noise environments given, there is a danger it will generalize poorly to new noise environments. We thus evaluate the DRDAE model on test set B which contains four noise environments not seen during denoising training, Table 2 shows the result.

Our model significantly outperforms the MFCC baseline, suggesting on unseen noise it still removes a substantial amount of noise from utterances. However, the DRDAE performs substantially worse than it did on test set A, which is to be expected as it trained on only four noise environments. For test sets A and B, the AFE performs similarly because its processing pipeline encodes no special knowledge of the noise environments. This

SNR	MFCC	AFE	DRDAE
Clean	0.94	**0.77**	0.94
20dB	7.94	**1.73**	2.24
15dB	20.51	**3.31**	3.87
10dB	43.81	**6.45**	8.61
5dB	70.04	**15.70**	21.39
0dB	86.29	**37.35**	51.25
-5dB	91.13	**69.99**	82.96

Table 2: Word error rates (WER) on Aurora2 test set B. Performance is averaged across the four noise environments in the test set. The test noise environments do not match the four noise environments used to train the DRDAE.

consistent performance across noise environments for the AFE leads it to outperform the DRDAE in this test condition. We note that the reported WER average of 12.25% for the SPLICE algorithm outperforms the DRDAE average of 17.47%. We hypothesize SPLICE generalizes better because separate models are trained for different noise environments and at test time the noise is matched to the most similar model. Our model makes fewer assumptions and is thus much more dependent upon the training set to provide a reasonable sample of noise environments that could be encountered at test time.

3.4. Denoising Model Comparison

Because of the many possible neural network architectures for denoising, we wish to explore what aspects of the architecture are important for good performance. This also serves to help understand whether architecture choice impacts how well the model generalizes to unseen noise conditions. We thus train versions of the model which are shallow as opposed to deep, and non-recurrent. Because the WER performance metric additionally depends upon HMM training, we compare models with a mean-squared error (MSE) metric to directly measure how well they predict clean data $\mathbf{y}$ from noisy input $\mathbf{x}$.

We are interested in both how well the models fit the training data, and generalize to a type of noise unseen during training. For this, we train the models using three of the four noise types present in the multi-condition training set, and measure performance on the fourth noise type as a development set. Clean utterances of the Aurora2 test sets are not readily available so we do not evaluate MSE on the test sets.

We train single hidden layer recurrent (RDAE) and non-recurrent (DAE) denoising models with 1000 hidden units each. We also train the same 3 layer with 500 hidden units each DRDAE model as used in the ASR experiment. Additionally, we train a non-recurrent version of this model (DDAE). All models are trained with the same input window size and on the same training set. Table 3 shows the training and development set MSE results, as well as the MSE of the corrupted input for each noise condition. We see that both temporally recurrent connections and multiple hidden layers are essential in both fitting the training set well and generalizing to unseen noise types. Furter work must explore increasing the model and training set sizes to learn denoising functions which better generalize to unseen noise environments.

	Input	DAE	RDAE	DDAE	DRDAE
Clean T	0	61.20	8.70	34.57	**7.49**
20dB T	47.13	69.37	30.68	43.07	**30.16**
15dB T	52.47	70.49	33.05	45.28	**32.42**
10dB T	58.52	72.96	36.12	48.95	**35.34**
5dB T	64.91	76.65	40.11	53.98	**39.17**
Clean D	0	61.27	8.76	34.64	**7.55**
20dB D	50.20	65.61	**34.42**	47.83	34.73
15dB D	56.05	67.85	**37.40**	50.94	37.55
10dB D	60.77	69.54	39.92	53.75	**39.69**
5dB D	66.06	74.02	43.71	58.50	**43.26**

Table 3: Average mean squared error (MSE) of denoised input with respect to the true clean features. One noise type from the Aurora 2 multi-condition training set (T) was used as a development set (D) to assess how well models generalize to an unseen noise type. One and two layer models were trained with and without recurrent connections for comparison. The MSE of the noisy input serves as a reference for the error metric.

4. References

[1] ETSI, "Advanced front-end feature extraction algorithm," *Technical Report. ETSI ES 202 050*, 2007.

[2] A. Ragni and M. Gales, "Derivative Kernels for Noise Robust ASR," in *ASRU*, 2011.

[3] S. Tamura and A. Waibel, "Noise reduction using connectionist models," in *ICASSP*, 1988, pp. 553–556.

[4] S. Parveen and P. Green, "Speech recognition with missing data using recurrent neural nets," in *NIPS*, 2001.

[5] G. Dahl, D. Yu, and L. Deng, "Large vocabulary continuous speech recognition with context-dependent DBN-HMMs," in *Proc. ICASSP*, 2011.

[6] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. Huang, "High-performance robust speech recognition using stereo training data," in *ICASSP*, 2001.

[7] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML*. ACM, 2008, pp. 1096–1103.

[8] Q. V. Le, A. Coates, B. Prochnow, and A. Y. Ng, "On Optimization Methods for Deep Learning," 2011.