An Investigation of Deep Neural Networks for Noise Robust Speech Recognition

Mateusz Buda
buda@kth.se

Masoumeh P. Ghaemmaghami
mpg@kth.se

Lin Qi
linqi@kth.se

Abstract

We investigate two efficient approaches for having robust speech recognition by using deep neural network and evaluate noise robustness in the proposed methods. In the first method, we improve robustness in conventional GMM-HMM recognizers by training on multi-condition data. It means that the system is trained with clean and also noisy data with different noise type and signal to noise ratios (SNR). The second method uses dropout to train a network that is more robust in terms of noise variance. Aurora 4 database is used for our experiments. It was created by artificially adding different types of noises to the Tidigits speech dataset. Experimental results show that generalization with dropout achieves significant improvement.

1 Introduction

Speech recognition is today a quite applicable technique in our lives. Telephone services for banks or hospitals, cellular phones, and many more products use speech recognition. The speech recognition systems performance is perfect in noiseless environments; but our world is full of different kinds of noise which reduce the level of recognition rate. HMM-based model of speech recognition system with Gaussian mixture model (GMM) for each state is sensitive to the mismatch between training and the operating conditions with environmental noises. Several approaches have been developed to reduce this mismatch. They can be classified in three categories [1].

The first category is speech enhancement approaches. In these approaches, before extraction of the relevant features, the effect of the noise on the speech signal is modeled and the distortion of the noise is reduced.

The second category is robust feature extraction. In this category the features of the speech signal are designed to be less sensitive to the noisy conditions. This is related to feature extraction methods which reduce the influence of noise [2].

The third category is model compensation techniques. Model compensation approaches determine the effect of the noise on the distributions of the speech features and adjust the models used in the recognition to reduce the effect of the noise. In this category, the model should be retrained with new noisy data for different environments. But, in the first and second category the model is trained with clean data and the noise effect is removed in the feature extraction stage. Both of these approaches could be more improved by using the multi-condition training data and adaptive training techniques [3]. Using the combination of feature enhancement or model adaptation with adaptive training become the most effective methods for robust speech recognition [4], [5], [6].

Recently, deep neural networks (DNN) are used as a new form of acoustic model. These acoustic models are similar to the original ANN-HMM hybrid architecture [7] but there are two key differences. The first difference is related to prediction of context-dependent acoustic states called senones [3]. Second difference is that, DNN has more layers than the networks trained in the past. Previous work in deep neural networks focused on generating posterior features [8], [9] or feature enhancement methods which map from noisy to clean features [10], [11].

In this paper, we focus on the noise robustness performance of DNN-based acoustic models and evaluate two methods to improve accuracy of recognition rate. In the first method, DNN is used in feature space and also model-space noise-adaptive training is done. This method uses information about the environmental distortion before network training. The second approach is to use dropout, which is suitable to prevent overfitting and produces a network that is more robust to variabilities in the input.

The results of our experiments on the Aurora 4 show that the DNN acoustic model has significant noise robustness, with high performance comparing to other complicated methods and the performance of speech recognition improved considerably in noisy environment. Also the tested methods unlike other robustness techniques for GMM-HMM acoustic models, don't add any complexity to the system [12].

The rest of the paper is organized as follows. In section 2, two strategies to improve noise robustness are presented. The performance of the proposed approaches is evaluated in section 3. Results are presented in Section 4. Conclusions are in section 5.

2 Method

Deep neural network (DNN) is simply a multi-layer perceptron (MLP) with many hidden layers. In this section we shortly review some problems with training DNNs and solutions to them. Next we describe how DNN can be used as an acoustic model for speech recognition and describe methods that we evaluated.

2.1 Deep neural networks

Starting with randomly initialized weights of DNN, back propagation training is very prone to finish in a local minimum, especially as the number of layers increases. To avoid this, we can pre-train the model to have better initialization before we start back propagation. One of the widely used methods of pre-training is growing the network layer by layer in an unsupervised manner. It is achieved by training pair of layers as a restricted Boltzmann machines (RBM) and then stacking them to make a DNN. RBMs are trained using an contrastive divergence objective criterion [13].

Another big problem with training DNNs is called vanishing gradient. The weights of units in those layers that are far from the output where the error is reported, are barely updated during the backward pass of back propagation. In short, it is caused by the fact that the maximum value of sigmoid function derivative equals 0.25. The most simple solution is to replace sigmoid function with rectified linear unit (ReLU) defined as

$$f(x) = max(0, x).$$

ReLU has derivative equal to 1 for $x > 0$ and 0 for $x < 0$ [14].

Applying DNN to speech recognition is done by replacing the state emission likelihoods generated by GMMs with likelihoods generated by the DNN. However, DNN outputs posterior probabilities of the form $p(s|x)$ that needs to be converted to the likelihood $p(x|s)$ required by the HMM framework. It can be done by dividing posterior probabilities by the frequencies of the HMM-states in the forced alignment that is used for DNN training [15].

2.2 Training with multi-condition speech

Training with multi-condition speech means that the training set contains data with different type of noise and SNR. This strategy should enable the network to learn higher level features that are more invariant to the effect of noise. The earlier layers are supposed to seek for features that are invariant across various acoustic conditions present in the training data [3].

We achieve this in DNN by training it with the input vector $\boldsymbol{v}_t$ that is extended context window of the noisy observations.

$$\boldsymbol{v}_t = [\boldsymbol{y}_{t-\tau}, \ldots, \boldsymbol{y}_{t-1}, \boldsymbol{y}_t, \boldsymbol{y}_{t+1}, \ldots, \boldsymbol{y}_{t+\tau}]$$

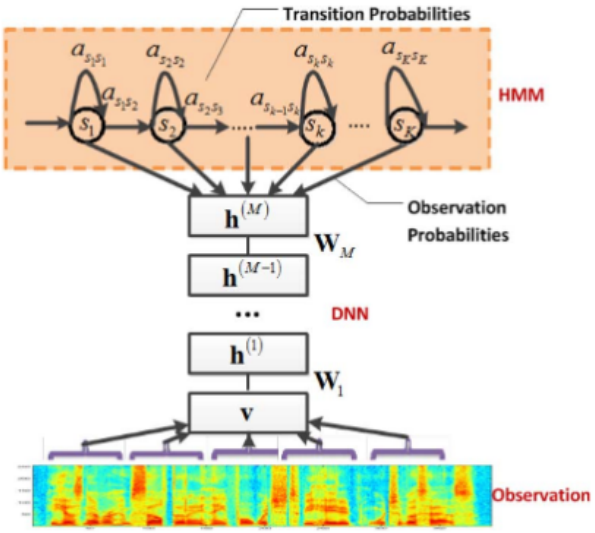

Figure 1: Diagram of hybrid architecture employing a deep neural network. The HMM models the sequential property of the speech signal, and the DNN models the scaled observation likelihoods. (Figure from [16].)

2.3 DNN dropout training

One of the biggest problems in training DNNs is overfitting, because the window size is bigger than the time shift, so when a large DNN is trained by a relatively small training set, overfitting will become noticeable. A training method called "dropout" is used here to alleviate this problem.

The basic idea for dropout method is to randomly omit a certain percentage (e.g. α) of the neurons in each hidden layer during each presentation of the samples during training [17]. This means that some of the neurons in the hidden layer will not work, but still keep their weights (because they may be used in the next training pass). Because of this, each random combination of the remaining hidden neurons must still perform well, which means that each neurons has little dependence on the others.

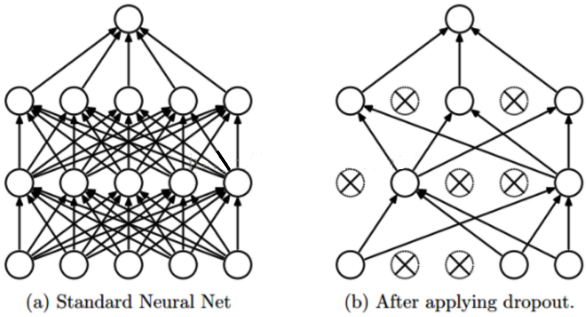

Figure 2: Basic DNN and DNN after using dropout function

The basic implement for the dropout is shown in the following equations. As we know the general DNN's equation is:

$$z_i^{(l+1)} = w_i^{(l+1)} * Y^l + b_i^{(l+1)} \tag{1}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \tag{2}$$

where l is the layer, Y is the last layer's neurons and b is the random noise.

After using dropout, the equations will be changed to:

$$r_j^l \sim Bernoulli(p) \tag{3}$$

$$y*^{l} = r^{l} * Y^{l} \tag{4}$$

$$z_i^{(l+1)} = w_i^{(l+1)} * y*^{l} + b_i^{(l+1)} \tag{5}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \tag{6}$$

As we see we use Bernoulli equations to generate a 0,1 vector randomly.

Moreover, dropout is a good way to insert random noise into training data, because each higher layer neurons get input from a random collection of lower layer neurons.

In training time, dropout essentially reduces the capacity of the DNN and thus can improve the generalization of the resulting model. Note that in [17] the dropout means that neurons' activation is set to 0 in order not to let an error signal pass through it. The same method is applied in the PDNN toolkit [18] that we use.

In test time, however, instead of using a random combination of the neurons at each hidden layer, the average of all the possible combinations is used. This can be easily accomplished by discounting all the weights involved in dropout training by $(1 - \alpha)$ and use the resulted model as a normal DNN. And also, in the test time, we can use dropout for all the n times, and get average result from all n times which is in order not to make x bigger and shrink the weights.

Dropout was successfully applied to TIMIT phoneme recognition in [17].

3 Experiments

3.1 Data set description

Aurora 4 database was used to verify the performance of the evaluated approach for DNN acoustic model. It contains speech data with the presence of additive noises and linear convolutional distortions [19]. The training set contains multi-condition speech data which is time-synchronized with the clean-condition data. One half of the utterances were recorded by the primary Sennheiser microphone while the other half was recorded using one of a secondary microphones. Both halves include a combination of clean speech from clean-condition training set and speech corrupted by one of four different noises (subway, babble, car, and exhibition hall) at 5 to 15 dB SNR.

Four test sets are defined. The first one (test set A) has the type of noise shared with training set but different SNRs (-5, 0, 20). For evaluating our methods in different noisy condition, the second set (test set B) is designed to have three different from training set noise types (restaurant, street and airport) at -5 to 20 dB SNR. The third (test set C) includes data corrupted by two different noises: subway and street, each with the same range of SNRs as in the second test set, filtered with Modified Intermediate Reference System (MIRS) characteristic. The forth (test set D) is more similar to the train set which includes data corrupted by three different noises: subway, bubble and car at -5 to 20 dB SNR. Also, training set and all of the test sets except test set 1, contain the clean data with 4004 utterances from 52 male and 52 female speakers. They are split equally into 4 subsets with 1001 utterances each, with all speakers being present only in one of them.

3.2 Features and tested models

To create the training set with aligned and labeled frames we used GMM-HMM system with 64 states and 16 Gaussians per state. The input features were 39-dimensional MFCC features (with first and second order delta features) with cepstral mean normalization.

We used two DNNs with identical architecture and trained them with exactly the same settings. The only difference between them was that one DNN has hidden layers with dropout factor of 20% and the other DNN that serves as a baseline does not use any dropout.

The size of input layer is 792 units for 24-dimensional log mel filterbank (FBANK) features with first and second order derivatives and context of 5 frames from each side. The output layer consists of 64 units corresponding to phoneme-states. The number of hidden layers is 3, each one containing 2048 units. The activation function used is rectified linear unit (ReLU).

Networks were trained in two phases: training and finetuning. Firstly we trained networks with 15 epochs using back propagation with a constant learning rate equal to 0.1 and momentum 0.9. Then to

finetune the models, 15 more back propagation epochs with learning rate 0.004 and momentum 0.9 were performed. At each training stage mini-batch size for stochastic gradient descent (SGD) was 512.

After the training is done, we evaluate and compare two networks on the 4 test sets defined in section 3.1. We measure a phoneme level classification accuracy of both networks to test the effect of using dropout. The network without dropout serves as a baseline in our tests.

4 Results

Figure 3 compares the training and validation error of both DNNs during the training and finetuning. It can be seen that the training error behaves very alike for both networks but the DNN with dropout scales better as the difference between validation and train loss is considerably smaller. Even though the baseline DNN reaches training error of 6% it performs much worse on validation set that barely improves during finetuning. The result is similar to the one obtained in [17] on the core test set of the TIMIT benchmark.

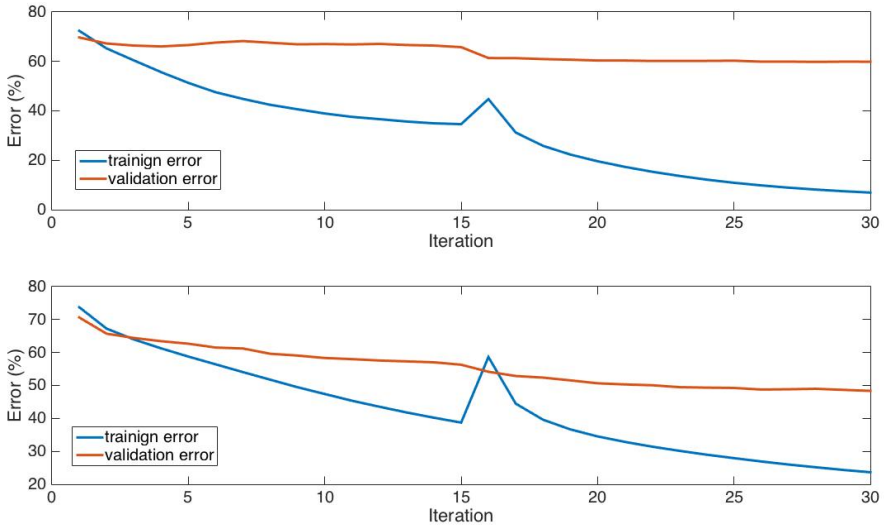

Figure 3: Comparison of training and validation errors for two tested DNNs. The upper row corresponds to the baseline DNN and lower one to the DNN with dropout.

The peak in training error at 16^{th} epoch for both networks comes from the fact that for finetuning (that starts from the 16^{th} epoch) we took only 3 hidden layers from the previous 15 epochs and the output layer was reinitialized. However, we also note at the same epoch a considerable decrease in validation error, again for both networks.

As shown in the table 1 and 2 that presents the accuracy of classification on each test set defined in section 3.1 dropout helped to increase the score on frame and phoneme level. The biggest improvement can be seen for test set B that contains noise of different type than the one we have in the training set. However, for all of them we observe better accuracy for model with dropout.

Table 1: Comparison of frame level error (%) on four test sets.

DNN type	A	B	C	D	AVG
DNN Baseline	63.15	69.39	62.63	62.02	64.30
DNN + Dropout	50.00	54.03	51.66	52.89	52.15

Table 2: Comparison of phoneme level error (%) on four test sets.

DNN type	A	B	C	D	AVG
DNN Baseline	54.40	58.92	55.03	54.16	55.63
DNN + Dropout	42.31	45.93	44.05	45.52	44.45

5 Discussion and Conclusions

As we expected, DNN with dropout significantly improved phoneme classification accuracy. During the learning neurons must have adapted to a situation when some input from a previous layer is missing and it helped to scale for previously unseen noise types in the test set.

Due to limited computational resources and memory needed to store the model in a GPU, we trained DNNs with only 3 hidden layers. As shown in [18] increasing the number of hidden layers to 5, 7 or even 9 gives even better results.

We also tested only one value of dropout factor (i.e. $\alpha = 20\%$), equal across all hidden layers. It needs to be further investigated what is the optimal value for dropout factor for robust speech recognition task and if it should be equal for all the hidden layers. Dropout can be also applied to the input layer but its value should be probably lower [20]. In [3] a method called noise-aware training was used together with dropout and performed best among all other tested.

References

[1] L.R. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall Signal Processing Series: Advanced monographs. PTR Prentice Hall, 1993.

[2] M. P. Ghaemmaghami, H. Sameti, Farbod Razzazi, B. BabaAli, and Saeed Dabbaghchian. using mlp neural network in log-spectral domain. In *2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 467–472, Dec 2009.

[3] M. L. Seltzer, D. Yu, and Y. Wang. An investigation of deep neural networks for noise robust speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7398–7402, May 2013.

[4] Tuomas Virtanen, Rita Singh, and Bhiksha Raj. *Techniques for noise robustness in automatic speech recognition*. John Wiley & Sons, 2012.

[5] O. Kalinli, M. L. Seltzer, J. Droppo, and A. Acero. Noise adaptive training for robust automatic speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):1889–1901, Nov 2010.

[6] H. Liao and M. J. F. Gales. Adaptive training with joint uncertainty decoding for robust recognition of noisy data. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 4, pages IV–389–IV–392, April 2007.

[7] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco. Connectionist probability estimators in hmm speech recognition. *IEEE Transactions on Speech and Audio Processing*, 2(1):161–174, Jan 1994.

[8] O. Vinyals and S. V. Ravuri. Comparing multilayer perceptron to deep belief network tandem features for robust asr. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4596–4599, May 2011.

[9] S. Sharma, D. Ellis, S. Kajarekar, P. Jain, and H. Hermansky. Feature extraction using non-linear transformation for on the aurora database. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, volume 2, pages II1117–II1120 vol.2, 2000.

[10] S. Tamura and A. Waibel. Noise reduction using connectionist models. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 553–556 vol.1, Apr 1988.

[11] Andrew 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 *INTERSPEECH*, 2012.

[12] Y. Tu, J. Du, Y. Xu, L. Dai, and C. H. Lee. Deep neural network based speech separation for. In *2014 12th International Conference on Signal Processing (ICSP)*, pages 532–536, Oct 2014.

[13] Geoffrey E. Hinton. A practical guide to training restricted boltzmann machines. In Grégoire Montavon, Genevieve B. Orr, and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade (2nd ed.)*, volume 7700 of *Lecture Notes in Computer Science*, pages 599–619. Springer, 2012.

[14] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. *Proc. ICML*, 30:1, 2013.

[15] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *Signal Processing Magazine*, 2012.

[16] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42, 2012.

[17] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[18] Yajie Miao. Kaldi+pdnn: Building dnn-based ASR systems with kaldi and PDNN. *CoRR*, abs/1401.6984, 2014.

[19] Jun Du, Qing Wang, Tian Gao, Yong Xu, Li-Rong Dai, and Chin-Hui Lee. Robust speech recognition with speech enhanced deep neural networks. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[20] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.