DIRECT MODELING OF RAW AUDIO WITH DNNS FOR WAKE WORD DETECTION

Kenichi Kumatani, *Sankaran Panchapagesan*, *Minhua Wu*, *Minjae Kim*[*],
Nikko Ström, *Gautam Tiwari*, *Arindam Mandal*
Amazon Inc.

ABSTRACT

In this work, we develop a technique for training features directly from the single-channel speech waveform in order to improve wake word (WW) detection performance. Conventional speech recognition systems typically extract a compact feature representation based on prior knowledge such as log-mel filter bank energy (LFBE). Such a feature is then used for training a deep neural network (DNN) acoustic model (AM). In contrast, we directly train the WW DNN AM from the single-channel audio data in a stage-wise manner. We first build a feature extraction DNN with a small hidden bottleneck layer, and train this bottleneck feature representation using the same multi-task cross-entropy objective function as we use to train our WW DNNs. Then, the WW classification DNN is trained with input bottleneck features, keeping the feature extraction layers fixed. Finally, the feature extraction and classification DNNs are combined and then jointly optimized. We show the effectiveness of this stage-wise training technique through a set of experiments on *real* beamformed far-field data. The experiment results show that the audio-input DNN provides significantly lower miss rates for a range of false alarm rates over the LFBE when a sufficient amount of training data is available, yielding approximately 12 % relative improvement in the area under the curve (AUC).

Index Terms— Keyword Spotting, Distant Speech, Raw Audio Input Acoustic Modeling

1. INTRODUCTION

Wake-word (WW) detection is the first important step before interactions through distant speech recognition [1–6]. WW detectors typically employ signal-processing techniques to obtain a compact feature representation such as LFBE [3–7] and tandem feature [8]. In this work, we develop the compact representation extraction method with DNNs directly from the raw audio.

A successful approach for audio-input DNN modeling from the single-channel audio would be using convolutional neural network (CNN) layers with long short-term memory (LSTM) [9] or network-in-network components [10]. The convolutional layers can model time-frequency characteristics of the single-channel audio frames very well, and then succeeding layers represent temporal characteristics of speech features.

In [11], a simple time-delay NN (TDNN) [12] model was trained on audio frames directly. The TDNN architecture used in [11] consists of a bottleneck DNN, followed by frame stacking and time derivatives of the bottleneck feature and several fully connected DNN layers. This TDNN architecture is also considered as a special case of 1-dimensional CNN layers with a filter stride equivalent to the frame shift, but without max-pooling. In their work, the bottleneck feature extractor was trained separately with the cross-entropy criterion. This can be suboptimal since it does not exploit the full potential of training directly on raw audio, i.e., jointly optimizing the feature extraction and classification layers, which would enable the model to learn more discriminative feature representations for the task at hand [9]. Accordingly, we develop a new training method to maximize the discrimination performance of both feature extraction and classification layers.

In the single-channel audio-input DNN work of [9–11], the respective models trained on raw audio provided results competitive or comparable to LFBE-input DNNs on ASR tasks, but not significantly better. Accuracy improvements are typically shown to obtain by a combination of the baseline LFBE-based and direct audio-based systems using feature or model combination.

Notice that this work only focuses on feature extraction on the single channel audio after acoustic beamforming [2, 13] in contrast to the multi-channel audio DNN work [14–16] where the beamforming and feature extraction layers are trained directly from the multi-channel data. We will show that a significant improvement can be achieved with the single channel data only although prior work [14–16] reported that there was no significant improvement with the single-channel input.

In this paper, we also propose a TDNN single-channel audio-input network for WW detection, similar in architecture to that in [11], but with some significant differences in the training procedure. Our feature extraction network computes bottleneck features using short 25ms audio frames as input, instead of 150ms frames as in [11]. The bottleneck features are then stacked over several frames, and input to the WW phone classification DNN. Moreover, we propose a new stage-wise training procedure for our audio-input WW DNNs, including joint optimization of the feature extraction and the classification DNNs. As it will be clear later, stage-wise training is shown to provide significantly improved WW detection accuracy using audio-input DNNs, compared to LFBE DNNs. We show the effectiveness of the jointly optimized audio DNN through the WW experiments on the real far-field data. We also investigate how much training data would be necessary to build the raw-audio input DNN in order to overcome the LFBE DNN. Moreover, we analyze the audio feature extraction DNN from the speech processing point of view.

The balance of the paper is organized as follows. Section 2 describes our baseline WW system with LFBE features, including the DNN-HMM based WW detection which is also used for audio-input DNNs. Section 3 proposes the new DNN training method on raw audio frames. In Section 4, WW detection results are presented. In Section 4, we investigate the effectiveness of the proposed stage-wise training, and also the sensitivity of WW performance with respect to the amount of training data. We also describe some analysis of the filter responses of the audio feature extraction DNN. Section 5 concludes this work.

[*] The author contributed to this work as an intern at Amazon, Sunnyvale.

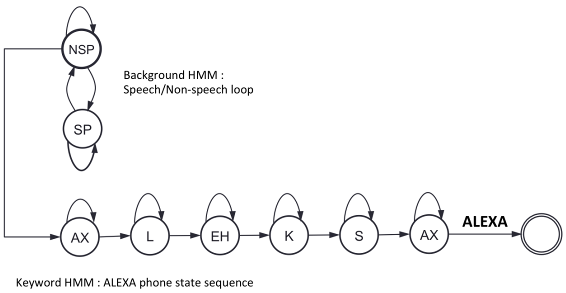


Fig. 1. HMM-based Keyword Spotting.

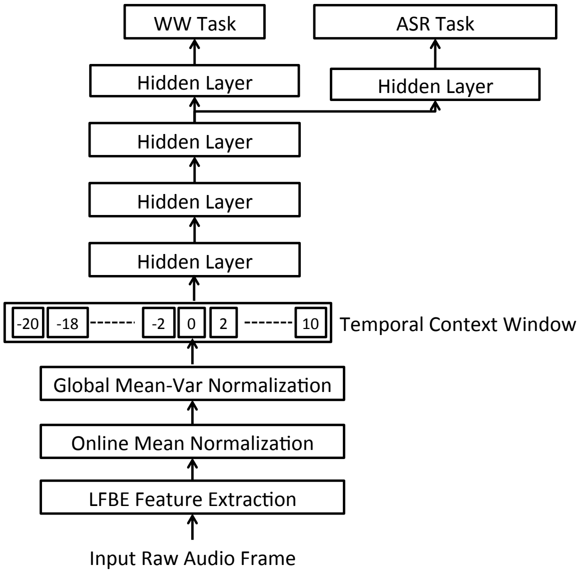


Fig. 2. Baseline WW DNN with the LFBE feature.

2. BASELINE WW SYSTEM

In this work, we employ the HMM-based approach with WW and filler/background HMMs [7]. Figure 1 illustrates an example of the finite state transducer (FST) at a phone level for a WW, with six phones in its pronunciation. Note that phone state is further divided into the HMM states. The HMM state is associated with a deep NN (DNN). The output layer of the WW DNN models the HMM states of the keyword(s) of interest (i.e., WW-specific phone state distributions), and the two 1-state background phones - speech and non-speech.

Figure 2 shows a schematic view of our DNN for the baseline WW system. Our baseline system first computes the LFBE feature from the enhanced speech [7]. In our system, audio is divided into overlapping frames of 25 milli-seconds (ms) with a frame shift of 10 ms. The LFBE features concatenated over multiple frames are then input to the phone classification DNN. The DNN consists of 3 layers of Affine transform and sigmoid activation components. In addition to those layers, we put two separate branches for WW and ASR tasks so as to classify WW-specific and context-dependent phones for our previously proposed multi-task training technique [7]. After the DNN is pre-trained layer-wise in a supervised fashion using a small subset of the training data, the entire DNN is further optimized with a distributed, asynchronous, stochastic gradient descent (SGD) training method [17].

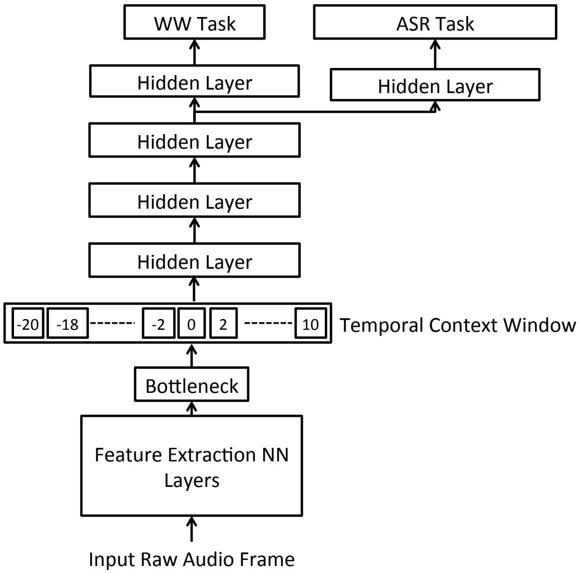


Fig. 3. Whole WW DNN from the audio input.

As illustrated in the FST of Figure 1, the WW hypothesis is generated when the final state of the WW FST is reached. We tune transition parameters and exit penalties in the WW and background HMMs for better accuracy, and a detection error tradeoff (DET) curve can be obtained by plotting the lowest achievable false alarm rate (FAR) at a given miss rate (MR) or false reject rate (FRR). The DET curves that will be described in Section 4 are obtained in this manner.

3. AUDIO-INPUT DNN

Figure 3 illustrates our entire DNN architecture with the direct audio input for WW. Figure 4 shows the details of feature extraction shown in Figure 3. The LFBE feature extraction block has been replaced with a bottleneck feature extraction DNN shown in Figure 4. With this bottleneck layer, we can efficiently reduce the dimension of a feature representation so as to incorporate the multiple frame contexts. As shown in Figure 4, the mean and variance of the raw audio signal are first globally normalized and passed through a set of affine transform and sigmoid activation layers, ending in a bottleneck layer which is the output of the feature extraction DNN. As shown in Figure 3, the bottleneck feature is spliced over several past and future frames through the temporal context window component whose output is fed into the hidden layers of the classification DNN. The WW classification DNN has output WW and ASR branches as with the baseline LFBE DNN.

In order to obtain a compact representation for WW phone discrimination, we first train the feature extraction DNN as shown in Figure 4. We wish that the bottleneck layer encodes a feature to

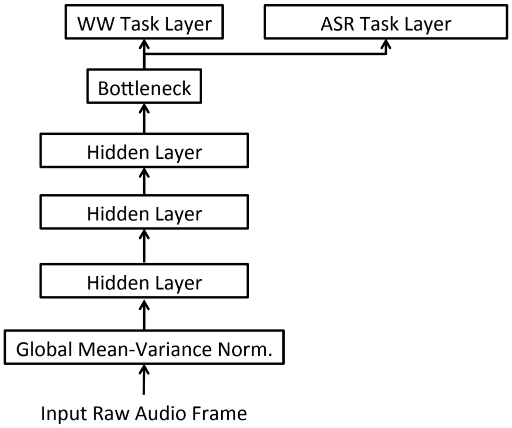


Fig. 4. Training the feature extraction DNN from raw audio input.

discriminate between the WW phones and background speech and non-speech sounds. The bottleneck layer is connected to soft-max layers for the WW and ASR tasks, and the feature extraction DNN is trained using the same multi-task cross-entropy objective function as our WW DNNs. This is considered the 1st stage in our stage-wise training procedure for audio-input WW DNNs. In our preliminary experiments, we found that this method of initializing the bottleneck feature provided better WW accuracy than an LFBE-imitation DNN trained based on L2 loss. We expect the feature extraction network will encode the parameters for speech processing, pre-emphasis, Hamming window and frequency transformation implicitly. Notice that the feature extraction DNN essentially functions as a 1-dimensional CNN with a stride of an analysis window shift since the weights of the feature extraction DNN are shared over multiple frames. Moreover, we also observed that the multiple-frame contexts of the bottleneck layer output led to better accuracy than a larger audio samples. We consider that the bottleneck layer can encode a feature representation more efficiently.

The entire network of the feature extraction and WW classification layers is jointly optimized based on the multi-task cross-entropy criterion. As we will discuss in Section 4, we empirically found that the initialization method for the entire DNN gave a big impact on the WW accuracy because of this deep architecture. Given the feature extraction DNN trained in stage 1 above, we propose two new initialization methods for the classification layers:

- Two-stage training, where the classification DNN is only pre-trained with a small portion of training data layer-wise before joint optimization as the 2nd stage.
- Three-stage training, where in the 2nd stage, the feature extraction DNN is fixed and the classification layers alone are trained on all the available data following layer-wise pre-training. This is followed by joint optimization of the feature extraction and classification DNNs as the 3rd stage.

As a reference, we will also compare these stage-wise training methods to the conventional pre-training method, adding layers after pre-training each with random initialization from the first feature extraction layer to the last classification layer. This would correspond to a 1-stage training procedure without any special initialization of the feature extraction DNN.

	LFBE DNN	Audio DNNl
Pre-emphasis	Yes	No
Window	Hamming	No
Online Feature Normalization	Yes	No
No. DNN Parameters	1.09 M	1.08 M

Table 1. Parameter configuration for experiments.

4. EXPERIMENT

Training data used here consist of several thousand hours of the real far-field data captured in various rooms. This contains approximately several hundred thousand subjects. In order to improve the robustness against noise unseen in the training data, the training data are artificially corrupted and the SNR is adjusted from 0 to 40 dB uniformly. For the test, we use two kinds of dataset, development and evaluation data set. Both of the development and evaluation data contain over several thousands of speech segments uttered by hundreds of subjects. The total duration of the development and test sets exceeds 60 hours. Those test sets cover noisy conditions and music playback cases. The captured far-field array data are processed with beamforming and acoustic echo cancellation [2, 13].

Table 1 contrasts the LFBE model setting to the audio models in feature extraction. We use almost the same number of weights for the LFBE and audio input DNNs. Thus, the audio input DNN saves the computation for entire feature computation as well as online feature normalization.

Here, we present results in the form of DET curves along with area under the curve (AUC) numbers, which will allow relative comparison of various systems in terms of performance impact. Note that since we present AUC numbers for DET curves instead of ROC curves, lower AUC numbers correspond to better performance. All the DET curves in this paper only show false alarm rates up to a multiplicative constant, due to the sensitive nature of this information. Thus, the DET curves presented here indicate the relative improvement or degradation. However, the plots and the AUC values still accurately preserve the relative performance improvements between different systems across a range of reasonable operating points.

4.1. Comparison of LFBE and raw Audio DNN

Figure 5 shows the DET curves obtained with the baseline LFBE and audio-input DNNs on the development data. In Figure 5, the classification layers of the audio-input DNN are trained in the stage-wise fashion described in Section 3. Again, as a reference, we also plot the DET curve of the audio-input DNN pre-trained from the first layer of the feature extraction DNN straight to the last layer without stage-wise training. That reference curve is labeled as Audio DNN with conventional pre-training in the figure. In order to generate the DET curves for Figure 5, we choose the best FST parameters with 4 HMM thresholds. Since we choose the FST parameters from the same pool of the FSTs, this result comparison is still fair. Those DET curves on the development data indicate the best possible WW performance without the grammatical language constraint. It is clear from Figure 5 that the audio-input DNNs trained stage-wise provide the better accuracy than the LFBE DNN. It is also clear Figure 5 that the best performance is obtained with the 3-stage trained audio DNN, which indicates that fully training the classification layer with the fixed feature extraction DNN gives good initial weight values. It is also clear by comparing the results of stage-wise trained audio DNN to that of the conventionally pre-trained audio DNN from Fig-

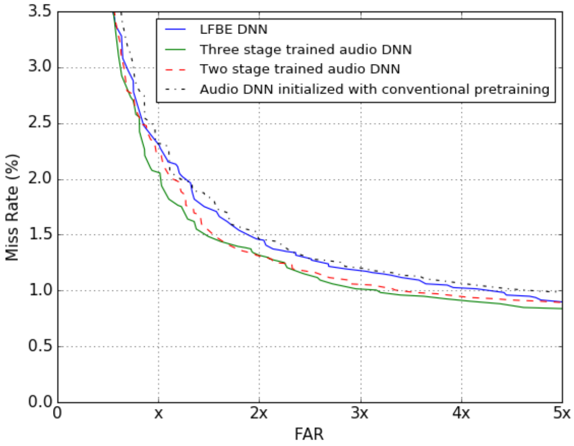


Fig. 5. DET curves of LFBE and audio DNNs on the development set.

WW Model Type	AUC	Relative Reduction
LFBE DNN	5.14 x	0 %
Audio DNN with conventional pre-training	5.29 x	-2.8 %
2-stage trained audio DNN	4.71 x	8.2 %
3-stage trained audio DNN	4.52 x	11.9 %

Table 2. AUCs calculated from the DET plots on the development set (Range of FARs used to compute AUC was $1x$ to $5x$ from Figure 5).

ure 5 that stage-wise training, training the feature extraction DNN separately, leads to the significantly better WW accuracy.

4.2. Effect with respect to an amount of training data

The only prior speech knowledge that we incorporate into the feature extraction DNN is phone class information defined by linguistics. In other words, the resultant DNN model may heavily depend on the training data and lack generalization. Accordingly, we investigate how much the size of training data gives an impact on the WW accuracy. Figure 6 shows the DET curves of the LFBE and audio-input DNN with respect to an amount of training data. It is apparent from Figure 6 that the audio-input DNN provides the better performance when the large amount of the training data is available. As expected, the performance of the LFBE DNN is saturated with a much smaller amount of training data (25% of all the available data). This result indicates that the raw audio DNN tends to require more data to generalize the WW model.

4.3. Effect of joint optimization

The proposed method can be viewed as an extension of the tandem or bottle-neck feature [8] to the hybrid DNN optimized jointly. An interesting question to be answered is how important it is to optimize the feature extraction and classification DNNs jointly. Figure 7 shows the DET curves that compare the DNN optimized jointly to the DNN without joint training. As a reference, the DET curve of the LFBE DNN with a comparative number of DNN parameters is described. It is clear from Figure 7 that joint optimization of the feature extraction and classification DNNs can improve WW accuracy. It is also clear from Figure 7 that the accuracy performance of the audio DNN is not as good as that of the LFBE without joint optimization.

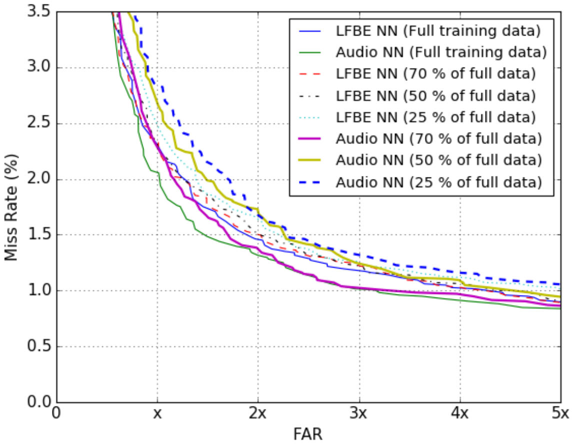


Fig. 6. DET curves on the development data set for different amounts of training data.

4.4. Results on Eval. Set without the optimum FST parameters

Here, we describe the final result on the evaluation set. In the previous figures, we choose the best FST parameters and HMM thresholds for each method in order to investigate the pure acoustic modeling performance for the WW task. We may not always know the best FST parameters. Accordingly, for the experiment on the evaluation set, we use the best FST parameters chosen from the development data set. Figure 8 show the DET curves on the evaluation set obtained with the audio and LFBE DNNs, given the FST parameters tuned from the development set. For the results on the evaluation set, we also obscure the absolute miss rate. However, again, the relative improvement is maintained precisely. Table 3 shows the AUCs on the evaluation set that correspond to Figure 8. It is clear from Figure 8 and Table 3 that the trend on the development set remains similar on the evaluation set regardless of the suboptimal FST parameters. The best performance was again achieved by the audio DNN trained in the three stage-wise manner even if the FST parameters are not optimum. This result verifies that the performance of the raw audio DNN is not influenced by the different FST parameters. The only different trend between the development and evaluation results is that the gap between the LFBE model and audio DNN initialized with conventional pre-training becomes bigger without help of the best FST parameters. This could be because the evaluation set has more noisy data.

4.5. Inside of the first linear transformation layer

In the case that raw audio samples are directly fed to the DNN, each row vector of the first linear transformation layer must form an FIR filter with an analysis window shift. Thus, analyzing such an FIR

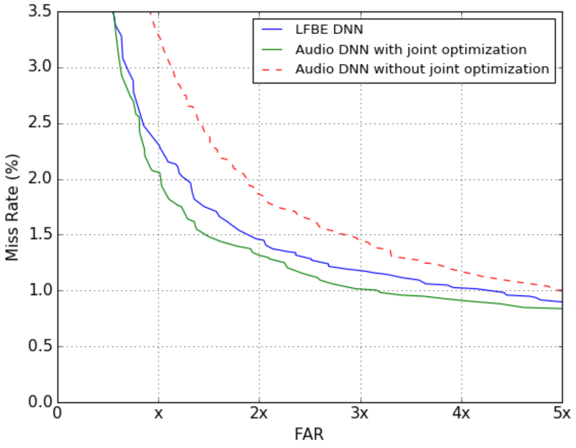


Fig. 7. DET curves on the development set: Effect of Joint optimization (Hybrid system).

WW Model Type	AUC	Relative Reduction
LFBE DNN	5.68 xy	0 %
Audio DNN with conventional pre-training	6.34 xy	-10.4 %
2-stage trained audio DNN	5.32 xy	6.3 %
3-stage trained audio DNN	4.96 xy	12.6 %

Table 3. AUCs of the DET plots on the evaluation set (Range of FARs used to compute AUC was $1.5x$ to $5x$ from Figure 8).

filter may shed light on what the DNN does on the raw audio samples. Figure 9 shows pairs of the FIR weights and frequency response samples obtained from the first linear transformation layer. In Figure 9, the left and right subfigures correspond to the row weight vector and frequency response, respectively. We can see from plots in Figure 9 that the first layer forms FIR filters so as to emphasize a certain range of frequency bands. Of course, this FIR filter is estimated solely from the training data so that the phones are well discriminated at each frame. Accordingly, we can hypothesize that the resultant filters attempt at extracting subbands necessary for discriminating phones.

5. CONCLUSIONS

In this paper, we have proposed the computationally efficient DNN architecture that takes the raw audio input. Through the WW experiments on the *real* far-field data, we showed that the audio-input NN outperformed the LFBE DNN in the case that the training data were sufficiently large. In the experiment, we observed that approximately 12 % of the AUC was reduced by the raw audio DNN system from the LFBE system. Moreover, we showed that building the feature extraction and classification DNNs in the stage-wise manner was important to achieve the better WW performance. Furthermore, we investigated the sensitivity of the performance as a function of the amount of training data. It turned out that the larger amount of training data was used, the better WW performance the raw audio DNN was able to provide than the LFBE DNN. Finally, we analyzed

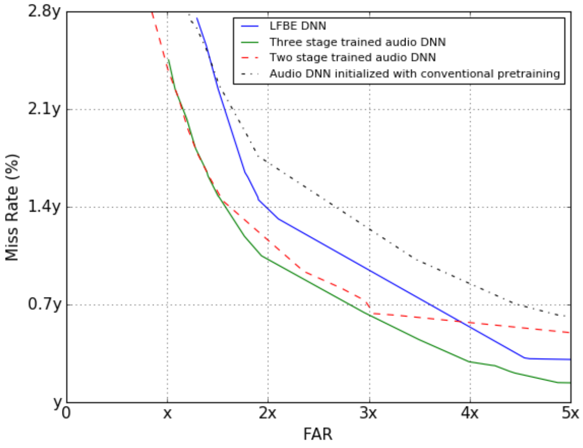


Fig. 8. DET curves of LFBE and audio DNNs on the evaluation set.

the inside of the audio feature extraction DNN. Analysis of the feature extraction DNN indicated that the input layer had an effect of subband frequency extraction and emphasis.

6. ACKNOWLEDGEMENT

We would like to you thank Aaron Challenner and Andrew Grasberger for supporting the implementation into the engine for the experiments.

7. REFERENCES

[1] K. Kumatani, J.W. McDonough, and Bhiksha Raj, "Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 127–140, 2012.

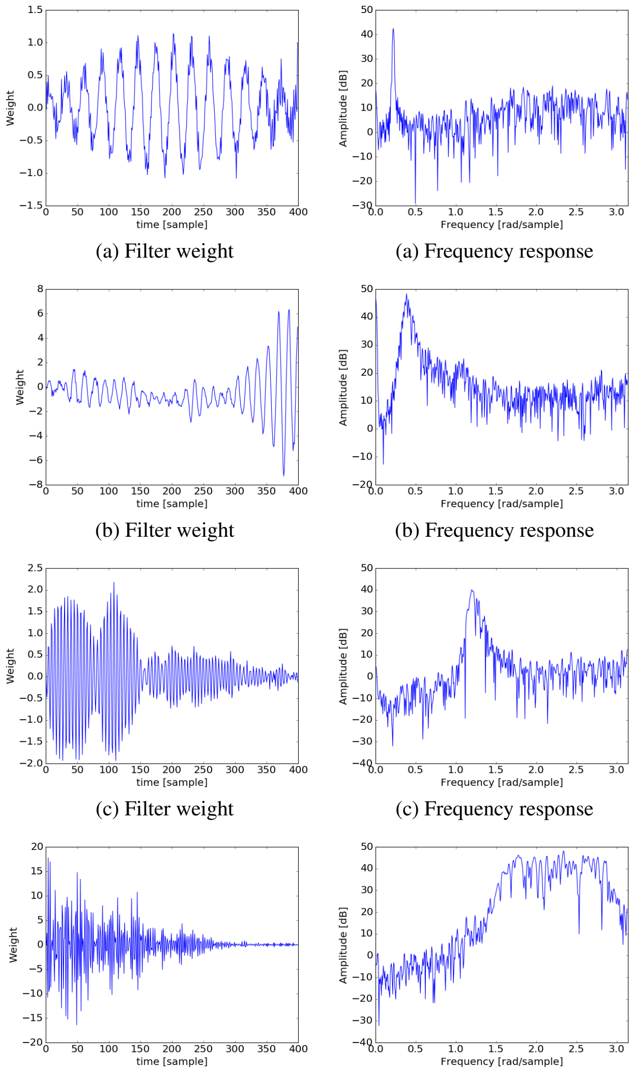
[2] Rohit Prasad, "Spoken Language Understanding for Amazon Echo," 2015, Keynote in Speech and Audio in the Northeast (SANE).

[3] M. Sun, V. Nagaraja, B. Hoffmeister, and S. Vitaladevuni, "Model shrinking for embedded keyword spotting," in *International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 369–374.

[4] M. Sun, A. Raju, G. Tucker, S. Panchapagesan, G. Fu, A. Mandal, S. Matsoukas, N. Ström, and S. Vitaladevuni, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," in *IEEE Spoken Language Technology Workshop (SLT) Workshop*, 2016.

[5] M. Sun, D. Snyder, Y. Gao, V. Nagaraja, M. Rodehorst, S. Panchapagesan, N. Ström, S. Matsoukas, and S. Vitaladevuni, "Compressed time delay neural network for small-footprint keyword spotting," in *Proc. Interspeech*, 2017, pp. 3607–3611.

[6] M. Sun, A. Schwarz, M. Wu, N. Ström, S. Matsoukas, and S. Vitaladevuni, "An empirical study of cross-lingual transfer learning techniques for small-footprint keyword spotting," in *International Conference on Machine Learning and Applications (ICMLA)*, 2017.



(a) Filter weight (a) Frequency response

(b) Filter weight (b) Frequency response

(c) Filter weight (c) Frequency response

(d) Filter weight (d) Frequency response

Fig. 9. Weights and frequency responses of the row weight vectors at the input layer

[7] Sankaran Panchapagesan, Ming Sun, Aparna Khare, Spyros Matsoukas, Arindam Mandal, Björn Hoffmeister, and Shiv Vitaladevuni, "Multi-task learning and weighted cross-entropy for dnn-based keyword spotting," in *Proc. Interspeech*, 2016, pp. 760–764.

[8] Nelson Morgan, "Deep and wide: Multiple layers in automatic speech recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 11, 2012.

[9] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Proc. Interspeech*, 2015.

[10] P. Ghahremani, V. Manohar, D. Povey, and S. Khudanpur, "Acoustic modelling from the signal domain using CNNs," in *Proc. Interspeech*, 2016.

[11] M. Bhargava and R. Rose, "Architectures for deep neural network based acoustic models defined over windowed speech waveforms," in *Proc. Interspeech*, 2015.

[12] Kevin J. Lang, Alex Waibel, and Geoffrey E. Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural Networks*, vol. 3, no. 1, pp. 23–43, 1990.

[13] M. Wölfel and J.W. McDonough, *Distant Speech Recognition*, Wiley, London, 2009.

[14] Ehsan Variani, Tara N. Sainath, Izhak Shafran, and Michiel Bacchiani, "Complex linear projection (CLP): A discriminative approach to joint feature extraction and acoustic modeling," in *Proc. Interspeech*, 2016.

[15] Tara N. Sainath, Arun Narayanan, Ron J. Weiss, Ehsan Variani, Kevin W. Wilson, Michiel Bacchiani, and Izhak Shafran, "Reducing the computational complexity of multimicrophone acoustic models with integrated feature extraction," in *Proc. Interspeech*, 2016.

[16] Bo Li, Tara N. Sainath, Ron J. Weiss, Kevin W. Wilson, and Michiel Bacchiani, "Neural network adaptive beamforming for robust multichannel speech recognition," in *Proc. Interspeech*, 2016.

[17] Nikko Ström, "Scalable distributed DNN training using commodity GPU cloud computing," in *Proc. Interspeech*, 2015, pp. 1488–1492.