CONVOLUTIONAL NETWORKS FOR SPEECH DETECTION

Somsak Sukittanon[1] *Arun C. Surendran*[2] *John C. Platt*[2] *Chris J. C. Burges*[2]

[1]University of Washington, Seattle WA 98195, ssukitta@ee.washington.edu
[2]Microsoft Research, Redmond WA 98052, {acsuren, jplatt, cburges}@microsoft.com

ABSTRACT

In this paper, we introduce a new framework for speech detection using convolutional networks. We propose a network architecture that can incorporate long and short-term temporal and spectral correlations of speech in the detection process. The proposed design is able to address many shortcomings of existing speech detectors in a unified new framework: First, it improves the robustness of the system to environmental variability while still being fast to evaluate. Second, it allows for a framework that is extendable to work under different time-scales for different applications. Finally, it is discriminative and produces reliable estimates of the probability of presence of speech in each frame for a wide variety of noise conditions. We propose that the inputs to the system be features that are measures of the true signal-to-noise ratio of a set of frequency bands of the signal. These can be easily and automatically generated by tracking the noise spectrum online. We present preliminary results on the AURORA database to demonstrate the effectiveness of the detector over conventional Gaussian detectors.

1. INTRODUCTION

There are several hurdles in designing good speech detection systems, first among which is the challenge of making it robust to changes in the environment or noise charecteristics. This problem is usually addressed in either the classifier or the feature extractor. The key is to design classifiers that have the ability to incorporate information about the signal itself, and can generalize well to unseen conditions without becoming too big or complicated. The key to feature design is to choose parameters that can be invariant to distortions, and can effectively incorporate short and long-term temporal and spectral correlations. Another challenge is to design systems that provide a good estimate of the probability of the presence of speech, rather than just a plain present/absent decision. Finally, it is always difficult to design detector frameworks that are flexible enough to be extendable across different time scales for different applications i.e. those that can be modified to detect speech either in a frame of 20ms (e.g. in a speech recognition application) or a segment of 100ms in length (e.g. for beamforming application).

In this paper, we introduce a speech detection framework based on convolutional networks that addresses all the above problems, primary among which is the robustness to changes in the environment. A convolutional network is a special type of a feedforward neural network which can incorporate prior knowledge about the signal and its distortions into its architecture. Thus it is a system that can do a joint feature-classifier design, and will be prepared to handle certain kind of distortions to the feature. In addition, it has the ability to use temporal and spectral correlations at different time-scales to make robust decisions, and can accurately estimate calibrated probabilities of the presence of speech in each segment. Finally, it is possible to use the same framework to train different detectors that operate on segments of different lengths. In this paper, we will only show results on frame level decisions. As an input to the system, we use signal-to-noise ratio based short-term spectral features. These features can be easily generated on-line. We will show that our new speech detector built on convolutional networks works well in an environment that is unseen and widely different from the conditions it was trained in.

The paper is organized as follows: We start Section 2 by briefly outlining some recent approaches to speech detection and highlight their shortcomings. We then describe a convolutional network and its properties, and how it can address many of the problems in designing a good speech dectector. In Section 3 we describe the features used as inputs to the system. Finally, in Section 4, we present results on the AURORA database that demonstrate the effectiveness of the proposed technique.

2. CONVOLUTIONAL NETWORKS FOR SPEECH DETECTION

2.1. Brief Look at Earlier Work

Earlier work in speech detection has focussed on individually addressing many of the problems mentioned in Section 1. Recently a speech detector using likelihood ratio (LR) tests based on Gaussian model was proposed [1]. The main advantage of this system over others was that it tracked the underlying the noise through a signal-to-noise ratio measure using a decision directed approach [1]. Though this technique was shown to be effective for speech detection under different noise conditions, it faces three main problems [2]. First, the LR scores do not translate easily into true class probabilities. Secondly, this method makes overly restrictive assumptions on the distributions of noise and speech spectra. The system uses short-term spectral features that are fragile in the presence of noise. Even if features that are robust to noise are derived externally, they may not work well with this system if their distributions do not match the Gaussian assumption. Thirdly, Gaussian models are especially poor when it comes to incorporating intra- and inter-frame correlations. In [1] inter-frame correlations were incorporated using HMM-like state transitions, but extending this idea beyond a single frame is complicated.

In [2] we presented a new system for speech detection using logistic discriminators that addressed some of these issues. This method has all the advantages of the Gaussian system, and in addition provides accurate estimates of posterior class probabilities for signals without making any assumptions on the underlying distributions. It is a discriminative detector, and is very simple and effective. But it has its own limitations: the system has too few parameters to take advantage of all the information in the signal, especially when the number of inputs become large. As a result, this framework is unable to scale to work at different time scales. Further, the feature-selection issue is not addressed by this classifier design. A more powerful generalized approach is to use neural networks for speech detection (e.g. [3]). Neural networks can easily learn complex non-linear mappings. When trained with the cross-entropy error functions, they are able to estimate true posterior class probabilities. The so-called *multi-conditioned training* i.e. training NNs using diverse data collected under various conditions, makes them work under unseen conditions. It can be argued that in a fully connected network with multiple hidden layers, some of the hidden layers can act as feature extractors. But when the input dimensions are high (e.g. when multiple frames of speech are used a input), the number of weights in a fully connected network becomes dramatically high and can lead to overfitting. What is needed is a solution that is a compromise between the simple classifiers that have problems learning, and complex classifiers that overfit.

2.2. Convolutional Networks

In this section we discuss the extension of feedforward networks to convolutional networks, and propose their application to speech detection.

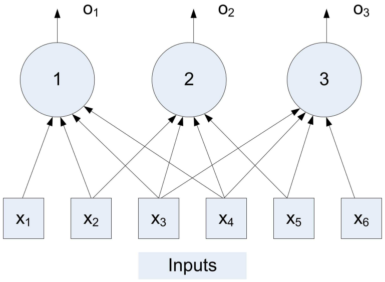

Fig. 1. Local receptors in one layer of a convolutional network.

Research in handwriting recognition in the past decade [4] has shown that it is possible to develop a specialized design for a neural network by incorporating prior knowledge into its architecture. This can be done in two ways: (1) restricting the network architecture by only using local connections, and (2) reduce the number of weights using weight sharing. This modification was motivated by the visual cortex of cats which had locally sensitive neurons that were also orientation selective [5]. For example, in Fig 1, each node is restricted to receiving inputs only from four nearest or "local" inputs. Further, if the weights w_1 through w_4 are restricted to be the same across all the hidden nodes, the output o_j of each node can be expressed as

$$o_j = f(\sum_{i=1}^{4} w_i * x_{i+j-1}), \qquad (1)$$

where w_is are the weights, x_{i+j-1} is the i^{th} input of node j, and $f(\cdot)$ is a non-linear squashing function like a sigmoid or tanh. The name *convolutional network* comes from the fact that the above equation is in the form of a convolution sum [5]. The weight vector can be thought of as as "kernel" which moves over the input performing local processing. One outcome of such a mechanism is that the outputs of the convolutional network become invariant to translation in the direction in which the weights are shared. For example, if the weight sharing is done for different inputs across time, the network is called a *time-delay neural network* (TDNN), and is invariant to shifting. Thus the mechanism of weight sharing can specifically incorporate the kind of distortion that can be expected. In fact, convolutional networks have been shown to recognize two-dimensional shapes with a high degree of invariance to various distortions like translation, scaling, skewing, etc [5, 4].

Further, the receptor fields acts as a kind of *local feature extractor*. In the visual cortex, for example, they extract features such as local edges, end-points, etc. In the TDNN, for example, the local receptors act as feature extractors in time. In the case of speech, depending upon the architecture, we can think of them as extracting time-frequency features. By having many such layers complicated feature extractors can be formed. In this sense, a convolutional network can be thought of as a joint feature extractor-classifier. We must mention that we don't think that these local receptor fields can completely replace the function of an external feature extractor e.g. the network cannot synthesize information that is not present in the data. The "noisiness" of the external features, and the amount of information they bear definitely affect the performance of the system. But the inherent ability of the system to create robust, learned internal representations is certainly one of the strengths of a convolutional network.

The parameters of the network are estimated during training using stochastic gradient descent by minimizing the cross-entropy error function:

$$\mathcal{E} = -\sum_{\mathcal{X}} t_{\mathcal{X}} \log(p_{\mathcal{X}}) + (1 - t_{\mathcal{X}}) \log(1 - p_{\mathcal{X}}), \qquad (2)$$

where $t_{\mathcal{X}}$s are the target labels for the training data $\mathcal{X}$. Thus the convolutional network is not only discriminative, but also provides the maximum likelihood estimate of the posterior class probability i.e. the probability that the given frame contains speech [6].

Couple of additional advantages of convolutional nets are: (1) the weight sharing not only achieves reduction in the number of parameters, but it also improves its ability to generalize by reducing its learning capacity [4]. Thus it is a good compromise between small networks that have difficulty learning and large networks that over fit. (2) The weight sharing makes it possible to implement the nework in parallel, unlike the traditional multilayer perceptrons.

2.3. Network Architecture

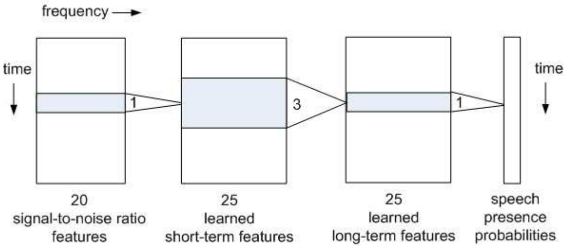

Fig. 2. Architecture of the convolutional network used in this paper.

We would like to design a convolutional network that can incoporate both spectral correlations in the short-term and long term-temporal correlations. One way to do this is to have two layers of feature extraction, each performing a specific function. Further discussions in this section will be done with reference to Figure 2. At each time index, the input to the first layer is a set of 20 features. These are measures of the SNR in each mel-frequency band in a 16ms interval (more about the features in the next section). The first layer has a "kernel" of 20 weights that operates on each time index independently to produce an output. Instead of having a single kernel, it is possible to have many kernels, each producing an independent non-linear representation of the input. These representations are called "feature maps". We generate 25 of these. Thus the first layer acts only on the data from one frame, and hence can be thought of as deriving a number of short-term features (the number of these features is equal to the number of feature maps).

We designed the second layer in the network to have a 1x3 kernel which acts on the outputs of all the feature maps from three time instants. This layer also has 25 feature maps. These can be interpreted as extracting longer-term temporal features from the data. The net effect is equivalent to using a window of three consecutive frames of prior SNR features to create one 20x3 input feature to the entire network. This window shifts forward in time by one frame at each time step. The size of the window is usually chosen based on the application and processing restrictions. For example, additive noise at one time instant does not affect too many future frames, so the window length can be fairly short. In a room with reverberation time of about 120ms, for example, a window length of 7 may be chosen. If a slight delay is allowed, frames from the near future may be included before a decision is made. If a delay in processing cannot be tolerated, only the current and past should be used.

Finally, the outputs layer has 2 nodes that are fully connected to the all the feature maps from the second layer with no weight sharing. The output of each node is the probability that the given input belongs to a certain class. If the other layers can be thought of as feature extractors, this layer can be thought of as a classifier. The net effect is to have a network that generates a set of probabilities at every time step, but needs three consecutive frames of input to generate this output. The network has a total of 2425 parameters. During training, the targets ($t_{\mathcal{X}}$ in Equation 2) for the output nodes are $\{1,0\}$ for all input data from speech segments, and is $\{0,1\}$ for all data from non-speech segments. The nodes in layers 1 and 2 use a tanh squashing funtion, while the output nodes use the softmax function.

3. PRIOR SIGNAL-TO-NOISE RATIO (SNR) BASED FEATURES

In [2] we used estimated *posterior signal-to-noise ratio* (SNR) based features: $\xi(k,t) = |Y(k,t)|^2/\hat{\lambda}(k,t)$, where $Y(k,t)$ is the spectrum of the input signal, $\hat{\lambda}(k,t)$ is the estimated noise energy and k,t are the frequency and time indices respectively. It is possible to derive an estimate of the actual signal-to-noise ratio (also called *prior* SNR) i.e. $\gamma(k,t) = |\hat{X}(k,t)|^2/\hat{\lambda}(k,t)$, where $\hat{X}(k,t)$ is the estimated spectrum of speech. $\gamma(k,t)$ can be derived in many ways. We follow the approach from [1]. First, the maximum likelihood estimate of the prior SNR is derived from the posterior SNR value using $\gamma(k,t)_{ML} = \xi(k,t) - 1$. Then this is smoothed from frame to frame using a decision directed approach [1]. $\gamma(k,t)$ seems to be slightly better than the posterior SNR for speech detection. To compute the spectrum we use a 128-pt modulated complex lapped transform (MCLTs) [7] every 16ms using a 32ms window. MCLT is a particular form of cosine modulated filter-bank that allows for perfect reconstruction. FFTs can easily be used instead of MCLTs without changing any other procedure in this paper.

Some preprocessing of the features is needed to improve generalization and learning accuracy. First, we convert the speech and noise spectrum into mel-band energies. This reduces the number of input parameters per frame and does not make any difference to the speech/non-speech detection accuracy [2]. Since short term spectra of speech are modeled well by log-normal distributions, we use the logarithm of the SNR estimate, rather than the SNR estimate itself. Then we normalize the input so that its variance is 1. In this paper, we precompute the variance for each coefficient over the training set and use it as the normalizing factor. The noise power λ is tracked automatically online using the method described in [2].

4. EXPERIMENTS AND RESULTS

4.1. Database

We use the well known AURORA database [8] for our experiments. The database has spoken digits from male and female speakers with different types of noises added to clean signals at 20, 15, 10 and 5dB SNR levels. We are interested only in two subsets of the database named TESTA and TESTB. The type of noise in TESTA and TESTB are different, though the speakers are the same. We chose 100 male and 100 female speaker data from TESTA for training the convolutional network; 10 male and 10 female speaker data from TESTA for validation during training, and 100 male and 100 female speaker data from TESTB for testing. We ensure that the speakers selected for testing from TESTB are entirely different from those in the training set. Also the data at 5dB SNR is not used for training. In total, about 55000 frames are available for training and over 68000 frames for testing.

The knowledge that this is a "stereo" database i.e. the data contains "clean" signals and their corresponding noisy counterparts, is used *only* to generate the true labels. This information was *not* used for online noise estimation or in testing. The true labels were generated by thresholding the energy in each frame of the clean data. The thresholds were selected so that all speech events were retained. This was verified through listening experiments on a small fraction of the training data. The threshold was tuned so that the low energy speech events and the transitions just barely made the cut.

4.2. Feature Comparison

The first set of experiments demonstrate the effectiveness of the prior SNR based features in comparison to the ones based on posterior SNR. Figure 3 shows the ROC curves for speech detection for two systems that are identical except for the features. The graphs show successful detection of speech frames vs. false alarms (non-speech frames classified as speech) for different SNRs. It is clear that the system using prior SNRs (solid line) in Figure 3 is better at all SNRs and in all parts of the curve.

4.3. Comparing Classifiers

In this section, we compare the performance of three different classifiers. We used a convolutional network with the architecture described in Figure 1, with 3 consecutive frames of input, 25 feature maps and 25 hidden nodes in layer 3. We experimented with different number of feature maps and hidden nodes, and the performances were similar. Figure 4 shows the ROC curves for

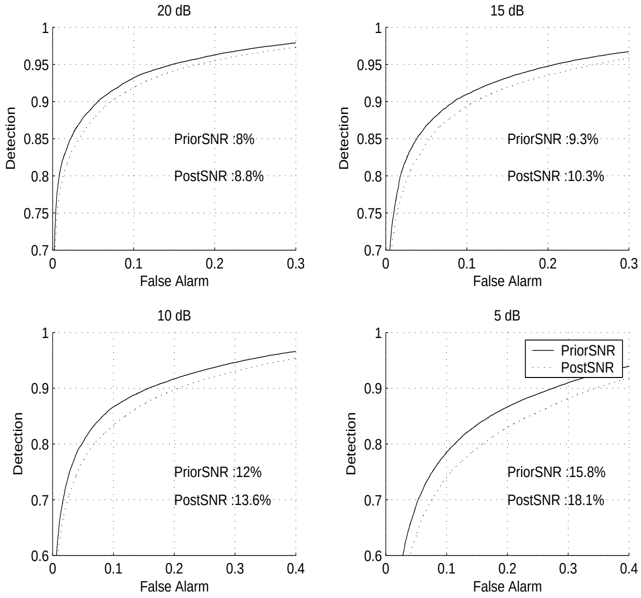

Fig. 3. ROC curves comparing features at various SNRs: (1) prior SNR based feature (solid line), (2) posterior SNR based features (dotted line). Minimum error numbers are inscribed.

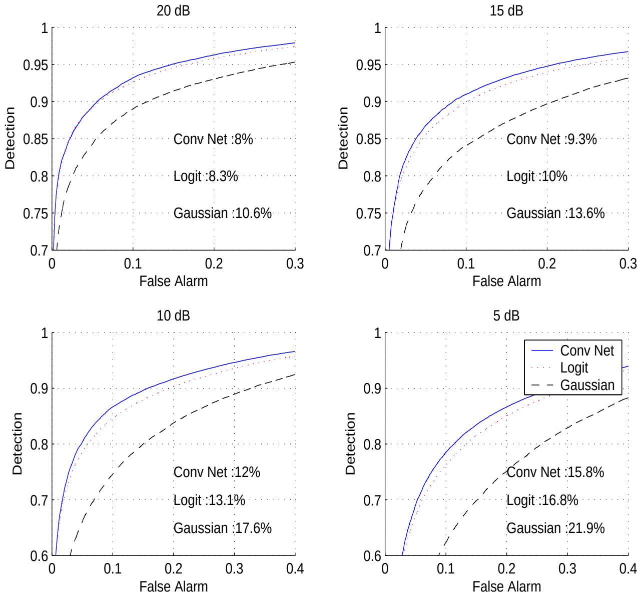

Fig. 4. ROC curves comparing the convolutional network to other dectectors at various SNRs. Results using convolutional network (solid line), logistic detector (dotted line) and the Gaussian based approach [1] (dashed line) are shown. Minimum error numbers are inscribed.

three systems: the convolutional network described in Section 2 (solid line), the logistic detector described in [2] (dotted line) and the Gaussian based approach [1] (dashed line). Minimum error numbers are inscribed in each graphs. The convolutional network outperforms the Gaussian method significantly. It improves the minimum error by approximately 25%, 32%, 32% and 23% under 20dB, 15dB, 10dB and 5dB conditions respectively. We should emphasize again that the network has not seen the speakers or the type of noise in the test set during training, and in fact did not train on any signal with 5dB SNR. The minimum error numbers do not tell the whole story - it is apparent from the curves that the proposed method outperforms the other methods in all parts of the ROC curve, and will be better than the Gaussian method for any application. The improvement of the new method over the logistic detector is small but consistent. Further, we experimented with the kernel width in the second layer. Using 3 frames reduced the minimum error by 12-14% for different SNRs over single frame inputs. Using more than a 3 frames did not significantly improve results with this noise set. It should be noted that the AURORA TESTB set mainly consists of additive noise. The new system may show significant difference compared to the logistic detector when used on data that has channel distortions and/or reverberation, especially when the number of input frames in increased.

5. SUMMARY

We present a new approach to speech detection using convolutional networks that is robust to changes in the environment. The convolutional network allowed us to incorporate prior knowledge about signals and distortions into its architecture. The major advantage of this that it improves the ability of the system to generalize to unseen environments without dramatically increasing the number of parameters. This also makes it possible to include both short-term and long term correlations without running into problems of scaling. We can train the network to provide good estimates of the probability that a given frame contains speech. We presented results on the AURORA database to demonstrate the effectiveness of the new approach.

6. REFERENCES

[1] J. Sohn et. al., “Statistical model based voice activity detector”, Sig. Proc. Letters, V.6, No.1, pp. 1-3, Jan ’99.

[2] A. C. Surendran, et. al., “Logistic Discriminators for Speech Detection Using Posterior SNRs”, accepted to ICASSP 2004, Montreal, available for download at http://research.microsoft.com/∼acsuren

[3] I. Shafran & R. Rose, “Robust Speech Detection and Segmentation for Real-time ASR Applications”, ICASSP 2003

[4] Y. Le Cun & Y. Bengio, “Convolutional Networks for Images, Speech, and Time-Series”, *Handbook of Brain Theory and Neural Nets*, pp 255-257. MIT Press.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Second Edition, 1999, Prentice-Hall.

[6] E. B. Baum & F. Wilczek, “Supervised Learning of Probability Distributions by Neural Networks”, NIPS-88, pp. 52-61.

[7] H. S. Malvar, “A modulated complex lapped transform and its applications to audio proc.”, ICASSP ’99, pp. 1421-1424.

[8] H. G. Hirsch, et. al., “The AURORA Experimental Framework”, ASR2000, Paris, France, Sep 2000.