RAW SPEECH SIGNAL-BASED CONTINUOUS SPEECH RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS

Dimitri Palaz Mathew Magimai.-Doss
Ronan Collobert

Idiap-RR-15-2014

OCTOBER 2014

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Raw Speech Signal-based Continuous Speech Recognition using Convolutional Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

State-of-the-art automatic speech recognition systems model the relationship between acoustic speech signal and phone classes in two stages, namely, extraction of spectral-based features based on prior knowledge followed by training of acoustic model, typically an artificial neural network (ANN). In a recent work, it was shown that convolution neural networks (CNNs) are capable of modeling the relation between acoustic speech signal and phone classes directly. This paper extends the CNN-based approach to large vocabulary speech recognition task. More precisely, we compare the CNN-based approach against the conventional ANN-based approach on Wall Street Journal corpus. Our studies show that the CNN-based approach with fewer parameters achieves performance comparable or better than the conventional ANN-based approach.

1 Introduction

State-of-the-art Automatic speech recognition (ASR) systems typically divide the task into several sub-tasks, which are optimized in an independent manner [1]. In a first step, the data is transformed into features, usually composed of a dimensionality reduction phase and an information selection phase, based on the task-specific knowledge of the phenomena. These two phases have been carefully hand-crafted, leading to state-of-the-art features such as mel frequency cepstral coefficients (MFCCs) [2] or perceptual linear prediction cepstral features (PLPs) [3]. In a second step, the likelihood of subword units such as, phonemes is estimated using generative models or discriminative models. In a final step, dynamic programming techniques are used to recognize the word sequence given the lexical and syntactical constraints.

Recent advances in machine learning have made possible systems that can be trained in an end-to-end manner, i.e. systems where every step is *learned* simultaneously, taking into account all the other steps and the final task of the whole system. It is typically referred to as *deep learning*, mainly because such architectures are usually composed of many layers (supposed to provide an increasing level of abstraction), compared to classical “shallow” systems. As opposed to “divide and conquer” approaches presented previously (where each step is independently optimized) deep learning approaches are often claimed to have the potential to lead to more optimal systems, and to have the advantage to alleviate the need of find the right features for a given task of interest. While there is a good success record of such approaches in the computer vision [4] or text processing fields [5], deep learning approaches for speech recognition still rely on spectral-based features such as MFCC [6]. Some systems have proposed to learn features from “intermediate” representation of speech, like mel filter bank energies and their temporal derivatives.

In a recent study [7], it was shown that it is possible to estimate phoneme class conditional probabilities by using raw speech signal as input to convolutional neural networks [8] (CNNs). On TIMIT phoneme recognition task, it was shown that the system is able to learn features from the raw speech signal, and yield performance similar or better than conventional ANN, more specifically multilayer perceptron (MLP), based system that takes cepstral features as input. The goal of the present paper is to ascertain that the findings on TIMIT phoneme recognition task scales to large vocabulary speech recognition task. Towards that end, we compare the CNN-based approach against the conventional ANN-based approach with different architectures on Wall Street Journal corpus. Our studies show that the CNN-based approach with fewer parameters yields similar or better performance than ANN-based approach.

The remainder of the paper is organized as follows. Section 2 presents a brief survey of related literature. Section 3 presents the classical HMM/ANN system. Section 4 presents the architecture of the proposed system. Section 5 presents the experimental setup and Section 6 presents the results. Section 7 presents the discussion and concludes the paper.

2 Related Work

Deep learning architectures have been successfully applied to a wide range of application: characters recognition [4], object recognition [9], natural language processing [10] or image classification [11]

In speech, one of the first phoneme recognition system based on neural network was the Time Delay Neural Network [12]. It was extended to isolated word recognition [13]. At the same time, hybrid HMM/ANN approach [14, 15] was developed, leading to more scalable systems.

Hybrid HMM/ANN approach was originally developed with ANNs that have single hidden layer and classify context-independent phonemes given cepstral feature as input. More recently, ANNs with deep learning architectures, more precisely, deep belief network or deep neural networks (DNNs) [16, 17], which can yield better system than a single hidden layer MLP have been proposed to address various aspects of acoustic modeling. More specifically, use of context-dependent phonemes [18, 19]; use of spectral features as opposed to cepstral features [6, 20]; learning features from raw speech signal and feeding them as input to ANN [21]; CNN-based system with mel filter bank energies as input [22, 23]; combination of different features [24]; CNN based phoneme recognition system with raw speech signal input trained in end-to-end manner [25]; multichannel processing using CNNs [26], to name a few.

3 Hybrid HMM/ANN system

As presented in Figure 1, hybrid HMM/ANN based ASR system is composed of three parts: features extraction, classification and decoding. In the first step, features are extracted from the signal, by transformation and filtering. The most common ones are short-term spectrum based features, namely, MFCCs or PLPs. Usually, the first and second derivative of these representations are computed over neighboring frames and used as additional features. They are then given as input to a Artificial Neural Network (ANN), along with some context, typically four frames of preceding and following context. The network is usually a feed-forward MLP composed of a hidden layer and an output layer, which estimates the conditional probabilities for each phoneme class. The MLP outputs are then used as emission probabilities or local score in a Viterbi decoder along with language model and phonetic dictionary to infer the output word sequence.

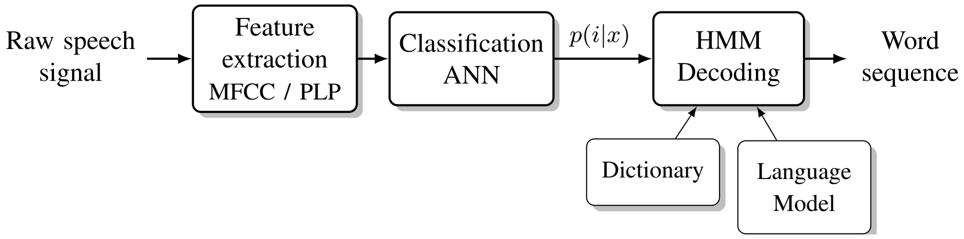

Figure 1: Hybrid HMM/ANN system. x here denotes cepstral features and i denotes a phoneme class.

4 Convolutional Neural Networks

In this paper, we use convolutional neural networks which replaces the first two stages of the hybrid system: the feature learning and the classification, as presented in Figure 2. The CNNs take raw speech signal as input, and output the conditional probabilities $p(i|x)$ for each class i, for each frame x. The two stages are trained jointly using the back-propagation algorithm.

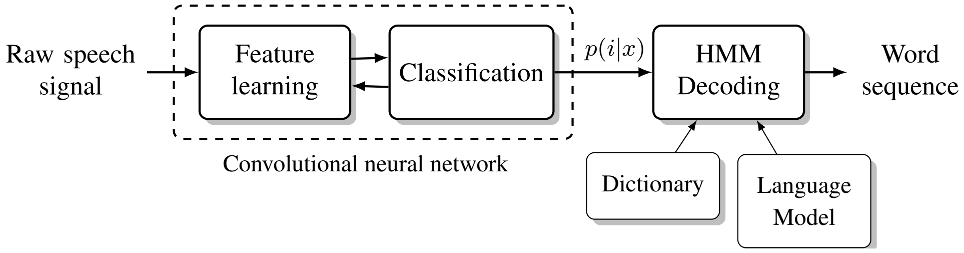

Figure 2: CNN-based ASR system. x denotes raw speech signal and i denotes a phoneme class.

4.1 Architecture

The network is given a sequence of raw input signal, split into frames, and outputs a score for each classes, for each frame. These type of network architectures are composed of several filter extraction stages, followed by a classification stage. A filter extraction stage involves a convolutional layer, followed by a temporal pooling layer and an non-linearity ($\tanh()$). Our optimal architecture included three stages of filter extraction (see Figure 3). Signal coming out of these stages are fed to a classification stage, which in our case is two linear layers with a large number of hidden units.

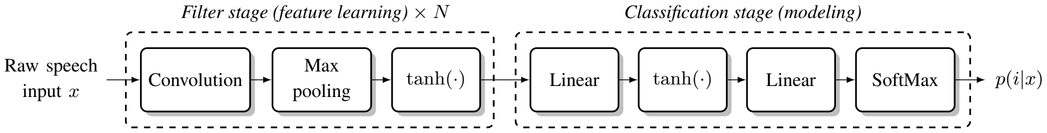

Figure 3: *Convolutional Neural Network. Several stages of convolution/pooling/tanh might be considered. Our network included 3 stages.*

4.2 Convolutional layer

While "classical" linear layers in standard MLPs accept a fixed-size input vector, a convolution layer is assumed to be fed with a sequence of T vectors/frames: $X = \{x^1 \quad x^2 \quad \dots \quad x^T\}$. A convolutional layer applies the same linear transformation over each successive (or interspaced by dW frames) windows of kW frames. For example, the transformation at frame t is formally written as:

$$M \begin{pmatrix} x^{t-(kW-1)/2} \\ \vdots \\ x^{t+(kW-1)/2} \end{pmatrix}, \tag{1}$$

where M is a $d_{out} \times d_{in}$ matrix of parameters. In other words, d_{out} filters (rows of the matrix M) are applied to the input sequence. An illustration is provided in Figure 4.

4.3 Max-pooling layer

These kind of layers perform local temporal max operations over an input sequence, as shown in Figure 5. More formally, the transformation at frame t is written as:

$$\max_{t-(kW-1)/2 \leq s \leq t+(kW-1)/2} x_s^d \qquad \forall d \tag{2}$$

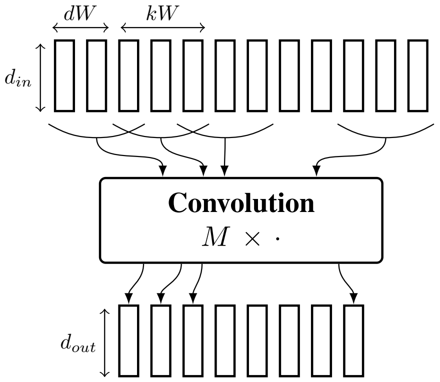

Figure 4: *Illustration of a convolutional layer. d_{in} and d_{out} are the dimensions of the input and output frames. kW is the kernel width (here $kW = 3$) and dW is the shift between two linear applications (here, $dW = 2$).*

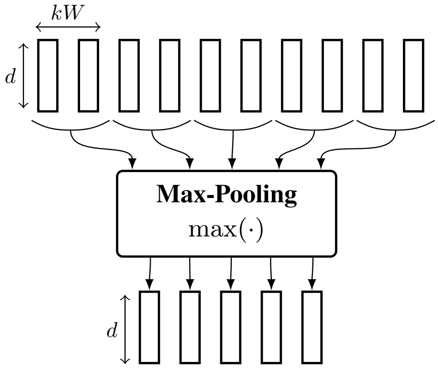

Figure 5: *Illustration of max-pooling layer. kW is the number of frame taken for each* max *operation (here, $kW = 2$) and d represents the dimension of input/output frames (which are equal).*

with x being the input and d the dimension. These layers increase the robustness of the network to slight temporal distortions in the input.

4.4 SoftMax layer

The $Softmax$ [27] layer interprets network output scores $f_i(x)$ as conditional probabilities, for each class label i:

$$p(i|x) = \frac{e^{f_i(x)}}{\sum_j e^{f_j(x)}} \tag{3}$$

4.5 Network training

The network parameters θ are learned by maximizing the log-likelihood L, given by:

$$L(\theta) = \sum_{n=1}^{N} \log(p(i_n|x_n, \theta)) \tag{4}$$

for each input x and label i, over the whole training set, with respect to the parameters of each layer of the network. Defining the `logadd` operation as: $\text{logadd}_i(z_i) = \log(\sum_i e^{z_i})$, the likelihood can be expressed as:

$$L = \log(p(i|x)) = f_i(x) - \underset{j}{\text{logadd}}(f_j(x)) \tag{5}$$

where $f_i(x)$ described the network score of input x and class i. Maximizing this likelihood is performed using the stochastic gradient ascent algorithm [28].

5 Experimental Setup

In this section, we present the setup used for the experiments, the database and the hyper-parameters of the networks.

5.1 Wall Street Journal Corpus

The SI-284 set of the Wall Street Journal (WSJ) corpus [29] is selected for the experiments. It is formed by combining data from WSJ0 and WSJ1 databases, sampled at 16 kHz. The set contains 36416 sequences, representing around 80 hours of speech. Ten percent of the set was taken as validation set. The Nov'92 set was selected as test set. It contains 330 sequences from 10 speakers. The dictionary was based on the CMU phoneme set, 40 context-independent phonemes, was used. 2776 tied-states were used in the experiment. They were derived by clustering context-dependent phones in HMM/GMM framework using decision tree state tying. The dictionary and the bigram language model provided by the corpus were used. The HMM/GMM system yields a performance of 5.1% word error rate. It is comparable to the performance reported in literature [30].

5.2 Features

Raw features are simply composed of a window of the speech signal (hence $d_{in} = 1$, for the first convolutional layer as shown in Figure 3). The window is normalized such that it has zero mean and unit variance. We also performed several baseline experiments, with MFCC as input features. They were computed (with HTK [31]) using a 25 ms Hamming window on the speech signal, with a shift of 10 ms. The signal is represented using 13th-order coefficients along with their first and second derivatives, computed on a 9 frames context.

5.3 Baseline systems

We compare our approach with the standard HMM/ANN system using cepstral features, as described in Section 2. We train ANNs with two different architectures. More precisely, MLP with single hidden layer referred to as ANN-1l and MLP with three hidden layers, referred to as ANN-3l. The input to the MLPs are 39 dimensional MFCC with several frames of preceding and following context. The number of context frame was tuned on the validation set. We do not pre-train the network.

5.4 Networks hyper-parameters

The hyper-parameters of the network are: the input window size, corresponding to the context taken along with each example, the number of sample for each example, the kernel width kW and shift dW of the convolutions, the number of filters d_{out}, the width of the hidden layer and the pooling width. They were tuned by early-stopping on the validation set. Ranges which were considered for the grid search are reported in Table 1. It is worth mentioning that for a given input window size over the raw signal, the size of the output of the filter extraction stage will strongly depend on the number of max-pooling layers, each of them dividing the output size of the filter stage by the chosen pooling kernel width. As a result, adding pooling layers *reduces* the input size of the classification stage, which in returns reduces the number of parameters of the network (as most parameters do lie in the classification stage).

The best performance was found with: 10 ms duration for each example, 310 ms of context, 9, 9 and 9 frames kernel width, 10, 1 and 1 frames shift, 80, 60 and 60 filters, 500 hidden units and 2 pooling width. For the baselines, the ANN-1l uses 500 nodes for the hidden layer and 21 frames as context. The ANN-3l system uses 1000 nodes for each hidden layer and 21 frames as context. The experiments were implemented using the *torch7* toolbox [32].

Table 1: Network hyper-parameters

Parameters	Range
Input window size (ms)	100-700
Example duration (ms)	5-15
Kernel width (kW)	1-11
Number of filters per kernel (d_{out})	20-100
Max-pooling kernel width	2-6
Number of hidden units in the class. stage	200-1500

5.5 Decoding

During decoding, the scaled likelihoods were estimated by dividing the posterior probability derived from the ANN or CNN by the prior probability of each class, estimated by counting on the training set. The hyper parameters such as, language scaling factor and the word insertion penalty were determined on the validation set.

6 Results

The results expressed in terms of Word Error Rate (WER) for the baseline systems and the proposed system are presented in Table 2, along with the number of parameters of the network. As it can be observed, CNN-based system outperforms the ANN-1l based baseline system . When compared to the ANN-3l baseline, the CNN-based system yields similar performance. These results suggest that CNNs result in simpler features which can be classified easily when compared to MFCC features.

Table 2: Word Error Rate on the Nov'92 testset

Feature	**Classifier**	**#Params.**	**WER**
MFCC	ANN-1l	1.8M	7.8 %
MFCC	ANN-3l	5.6M	6.4 %
RAW	CNN	3.1M	6.7 %

7 Conclusion

In this paper, we investigated the scalability of an ASR approach based on CNN, which takes as input the raw speech signal, to large vocabulary task. Our studies on WSJ corpus showed that the CNN-based system is able to achieve performance comparable to or better than the ANN-based system, which takes standard cepstral features as input. These findings are inline with the phoneme recognition studies reported on TIMIT corpus in [7]. Thus, indicating that the CNN-based approach is indeed scalable and can potentially remove the need to extract standard cepstral features. In order to further ascertain it, our future work will focus on studying the domain independence and language independence of the CNN-based approach, as the standard cepstral feature extraction process does not have any such dependency.

References

[1] H. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach.* Springer, 1994, vol. 247.

[2] 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*, vol. 28, no. 4, pp. 357–366, 1980.

[3] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, p. 1738, 1990.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[6] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14 –22, jan. 2012.

[7] D. Palaz, R. Collobert, and M. Magimai.-Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," in *Proc. of Interspeech*, Aug. 2013.

[8] Y. LeCun, "Generalization and network design strategies," in *Connectionism in Perspective*, R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels, Eds. Zurich, Switzerland: Elsevier, 1989.

[9] Y. LeCun, F. J. Huang, and L. Bottou, “Learning methods for generic object recognition with invariance to pose and lighting,” in *Proceedings of CVPR*, vol. 2, 2004, pp. II–97.

[10] R. Collobert and J. Weston, “A unified architecture for natural language processing: deep neural networks with multitask learning,” in *Proceedings of ICML*, 2008, pp. 160–167.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.

[12] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, “Phoneme recognition using time-delay neural networks,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 3, pp. 328 –339, mar 1989.

[13] L. Bottou, F. Fogelman Soulié, P. Blanchet, and J. S. Lienard, “Experiments with time delay networks and dynamic time warping for speaker independent isolated digit recognition,” in *Proceedings of EuroSpeech*, vol. 2, Paris, France, 1989, pp. 537–540.

[14] H. Bourlard and C. Wellekens, “Links between markov models and multilayer perceptrons,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 12, pp. 1167 –1178, Dec. 1990.

[15] Y. Bengio, “A connectionist approach to speech recognition,” *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 7, no. 4, pp. 647–668, 1993.

[16] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[17] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath, “Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups,” *Signal Processing Magazine, IEEE*, vol. 29, no. 6, p. 8297, 2012.

[18] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using context-dependent deep neural networks,” in *Proc. of Interspeech*, 2011, pp. 437–440.

[19] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, p. 3042, 2012.

[20] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, “Unsupervised feature learning for audio classification using convolutional deep belief networks,” in *Advances in Neural Information Processing Systems 22*, 2009, pp. 1096–1104.

[21] N. Jaitly and G. Hinton, “Learning a better representation of speech soundwaves using restricted boltzmann machines,” in *Proc. of ICASSP*, 2011, pp. 5884–5887.

[22] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, “Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition,” in *Proc. of ICASSP*, 2012, pp. 4277–4280.

[23] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, “Deep convolutional neural networks for lvcsr,” in *Proc. of ICASSP*, 2013, pp. 8614–8618.

[24] E. Bocchieri and D. Dimitriadis, “Investigating deep neural network based transforms of robust audio features for lvcsr,” in *Proc. of ICASSP*, 2013, pp. 6709–6713.

[25] D. Palaz, R. Collobert, and M. Magimai. -Doss, “End-to-end Phoneme Sequence Recognition using Convolutional Neural Networks,” *ArXiv e-prints*, Dec. 2013.

[26] P. Swietojanski, A. Ghoshal, and S. Renals, “Convolutional neural networks for distant speech recognition,” *Signal Processing Letters, IEEE*, vol. 21, no. 9, pp. 1120–1124, September 2014.

[27] J. Bridle, “Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition,” in *Neuro-computing: Algorithms, Architectures and Applications*, 1990, pp. 227–236.

[28] L. Bottou, “Stochastic gradient learning in neural networks,” in *Proceedings of Neuro-Nmes 91*. Nimes, France: EC2, 1991.

[29] P. Woodland, J. Odell, V. Valtchev, and S. Young, “Large vocabulary continuous speech recognition using htk,” in *Proc. of ICASSP*, vol. ii, apr 1994, pp. II/125 –II/128 vol.2.

[30] W. Reichl and W. Chou, “Robust decision tree state tying for continuous speech recognition,” *IEEE Transactions on Speech and Audio Processing*, pp. 555–566, 2000.

[31] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, “The htk book,” *Cambridge University Engineering Department*, vol. 3, 2002.

[32] R. Collobert, K. Kavukcuoglu, and C. Farabet, “Torch7: A matlab-like environment for machine learning,” in *BigLearn, NIPS Workshop*, 2011.