Very Deep Convolutional Neural Networks for LVCSR

Mengxiao Bi, Yanmin Qian, Kai Yu

Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{sxkachilles, yanminqian, kai.yu}@sjtu.edu.cn

Abstract

Convolutional Neural Networks (CNNs) have achieved state-of-the-art performance when embedded in large vocabulary continuous speech recognition (LVCSR) systems due to its capability of modeling local correlations and reducing translational variations. In all previous related works for ASR, only up to two convolutional layers are employed. In light of the recent success of very deep CNNs in image classification, it is of interest to investigate the deep structure of CNNs for speech recognition in detail. In contrast to image classification, the dimensionality of the speech feature, the span size of input feature and the relationship between temporal and spectral domain are new factors to consider while designing very deep CNNs. In this work, very deep CNNs are introduced for LVCSR task, by extending depth of convolutional layers up to ten. The contribution of this work is two-fold: performance improvement of very deep CNNs is investigated under different configurations; further, a better way to perform convolution operations on temporal dimension is proposed. Experiments showed that very deep CNNs offer a 8-12% relative improvement over baseline DNN system, and a 4-7% relative improvement over baseline CNN system, evaluated on both a 15-hr Callhome and a 51-hr Switchboard LVCSR tasks.

Index Terms: Very Deep Convolutional Networks, Speech Recognition, Acoustic Model, CNNs, Neural Networks

1. Introduction

Deep Neural Networks (DNNs), which have been applied in large vocabulary continuous speech recognition for years [1, 2, 3], gain significant improvement over state-of-the-art GMM-HMM systems. Due to its strong representative capability, DNNs can better predict state posteriors using given speech observations. However, DNNs are generic models, and are not deliberately designed to model input specialties, which in the case of speech are temporal and spectral local correlations, and small shifts of speech features influenced by speaker and environment variations [4, 5, 6].

Convolutional Neural Networks (CNNs), which can better model temporal and spectral local correlations and gain translational invariance, are first proved very effective in image classification task [7]. Recently, several different kinds of CNNs have been explored and achieved state-of-the-art performance in LVCSR [6, 8, 9, 10]. However, the complexity of CNNs leads to more variations of configurations. For example in [9], performance of the system using the configuration of 3 convolutional layers and 3 fully connected layers deteriorates compared to the configuration of 2 convolutional layers and 4 fully connected layers. However, it might be unfair to reach the conclusion that 2 convolutional layers is best for LVCSR in the same model parameter scale because the performance decline might be caused by not enough fully connected layers or convolutional layers of an insufficient depth.

Recently the computer vision community has found that image classification performance can be substantially improved using very deep CNNs. And surprisingly, the number of parameters remains the same magnitude or even smaller [11, 12], because convolution kernel sizes are carefully designed, for example 3×3, which is the smallest size to capture the notion of each direction in domain. Also, pooling layers are only applied after several convolutional layers.

In this work, we first explore the appropriate structure of very deep CNN for LVCSR on a 15-hr Callhome English task. Based on several fundamental setups of structure designing, we investigate the size of feature extension, the convolutional layer depth, ways of performing convolution operations on both dimensions, and whether convolutional depth has the most significant impact on performance. Eventually, our proposed very deep CNN obtained a 8-12% relative improvement over baseline hybrid DNN system, and a 4-7% relative improvement over baseline hybrid CNN system [9].

The rest of this paper is organized as follows. Section 2 gives a brief review of the typical CNN. In Section 3 fundamental setups for very deep CNNs and the configuration of the proposed very deep CNN are discussed and investigated. The experimental results on LVCSR tasks are presented in Section 4 and conclusions are given in Section 5.

2. A brief review of CNNs

A typical CNN has two major parts: a convolutional module followed by fully connected layers. In the convolutional module, there are two fundamental types of layers, namely convolutional layers and pooling layers. A convolutional layer does convolution operations to generate values from local regions (receptive fields) on channels of the previous layer. All neurons in one channel share the same filters affiliated with the channel. A pooling layer is simpler, which mainly downsamples channels of the previous layer, no matter what pooling strategy is used.

Considering one channel pair $\mathbf{h}^{(l-1)}$ and $\mathbf{h}^{(l)}$ in two adjacent layers, the convolution operation can be expressed as:

$$\mathbf{h}^{(l)} = \sigma\left(\mathbf{W}^{(l)} * \mathbf{h}^{(l-1)} + b^{(l)}\right) \quad (1)$$

where $\mathbf{W}^{(l)}$ is the filter applied on $\mathbf{h}^{(l-1)}$ to generate $\mathbf{h}^{(l)}$, $b^{(l)}$

This work was supported by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the China NSFC project No. 61222208 and JiangSu NSF project No. 201302060012.

is a bias scalar for channel $\mathbf{h}^{(l)}$, operation $*$ is the convolution operation, and activation function $\sigma(\cdot)$ can be sigmoid, hyperbolic tangent or rectified linear unit (ReLU). When multiple channels are present in the previous layer, all convolution results are simply summed before adding bias and then the nonlinearity function is applied.

Unlike the convolution operation, several kinds of pooling operations have been investigated [13], and the most popular pooling method is max-pooling: for each channel, the max value in each pooling window size is outputed.

Till now the most popular CNN configuration published for LVCSR is the configuration from [9], which is used as baseline CNN in this paper, listed as the last column in Table 1.

3. Architecture of very deep CNNs

Due to the complexity of very deep CNNs, it is impossible to try every possible configurations, some intuitive principles are used to determine fundamental setups for all experiments:

- Different from vision or image tasks, the size of neural network inputs, namely the speech feature dimension and the context window size, are comparatively smaller in most of the LVCSR systems[1]. Accordingly, the filter size, pooling size, convolutional output channel size and number of pooling operations have to be constrained to increase the convolutional depth.

  In more detail, convolution kernel size for both dimensions is set to 3 as it is the smallest size to capture local correlations on both temporal and spectral dimensions. Pooling size for both dimensions is set to 2, which is the smallest possible pooling size to reduce resolution of channels. Correspondingly, stride of convolution is set to 1 and stride of pooling is set to 2. In addition, convolutional output channel size is fixed to 1 × 3 to give space for feature extension, and consequently the convolutional layers could be seen as an independent module for the whole network. Furthermore, only 2 pooling operations are performed on both dimensions for most proposed structures to reserve size of channels for convolution operations.
- Next, the number of channels is well controlled that it starts at 64, doubles after two convolutional layers and one pooling layer, and doubles again after several layers depending on the model depth specifically. The number of channels increases gradually and reaches the maximum 256 in the end, which makes the number of parameters still comparable to baseline CNN. On the other hand, larger channel number also brings some side effects, such as longer training time and more memory usage.
- Finally, on top of the convolutional module a normal 4 × 2048 MLP is added followed by a softmax output layer. As shown in Table 1, seven different configurations $(\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \mathrm{B}, \mathrm{C}_1, \mathrm{C}_2, \mathrm{D})$ with 6, 6, 6, 8, 10, 10, 2 convolutional layers respectively are used in the following experiments, and model C_1 is illustrated in Fig. 1).

Based on above fundamental setups, all investigated configurations of CNNs in this work are listed in Table 1, where term M × N refers to a convolutional layer with corresponding filter size, and term [M × N] refers to a max-pooling layer with corresponding pooling size, where M and N denote temporal and spectral size respectively.

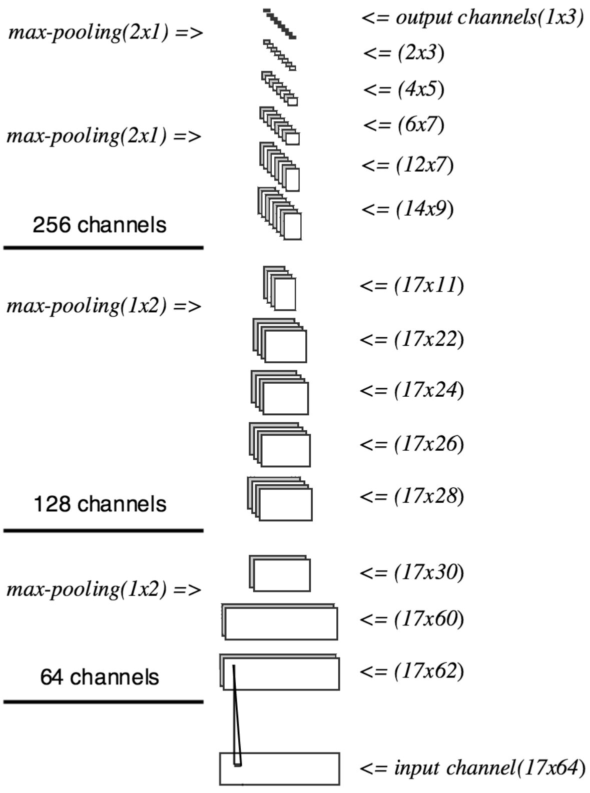

Figure 1: *Convolutional module of model C_1*

3.1. Experimental setup

To learn the behavior of very deep CNNs, different models are firstly evaluated on 15-hr Callhome English LVCSR task. A standard triphone GMM-HMM system with 1934 states is first trained using 13-dimension PLP features along with delta and double-delta (referred to as $\Delta/\Delta\Delta$ in rest of the paper), to obtain the alignment for NN training. 20 conversations in Callhome English corpus, about 2.0 hours, form the test set. A trigram language model obtained by interpolating individual language models trained on Callhome, Switchboard and Gigaword English corpus is used for decoding.

The baseline hybrid CNN system uses the same configuration as [9]: using the 40-dimension FBANK features with $\Delta/\Delta\Delta$ as inputs. The first convolutional layer has 128 channels and the second has 256. A 9 × 9 convolution is performed on the first convolutional layer and a 3 × 4 convolution is performed on the second. The first convolutional layer is followed by a size 3 max-pooling constrained on frequency and nothing is performed after the second convolutional layer. On top of the convolution part, a 4 × 2048 fully connected sigmoidal layers is applied to form the final model.

Kaldi [14] and PDNN [15] are used to train all neural networks here. RBM pre-training is used for DNN initialization and CNN is initialized randomly, and all neural networks are fine-tuned with cros-entropy criterion using stochastic gradient

[1] Normally using 11 × 40 inputs, 40-dim FBANK features with 11 consective frames.

Table 1: *Configuration details of different CNN structures, including the proposed very deep CNNs and the normal CNN.*

model	A_1	A_2	A_3	B	C_1	C_2	D	baseline CNN
input channel size	17×40	33×40	11×40	17×52	17×64	17×64	17×64	11×40
# of conv. layers	6	6	6	8	10	10	2	2
64 channels	1×3 1×3 $[1 \times 2]$	4×3 3×3 $[2 \times 2]$	1×3 1×3 $[1 \times 2]$	1×3 1×3 $[1 \times 2]$	1×3 1×3 $[1 \times 2]$	4×3 3×3 $[2 \times 2]$	-	-
128 channels	4×3 3×3 $[2 \times 2]$	3×3 3×3 $[2 \times 2]$	1×3 1×3 $[1 \times 2]$	1×3 1×3 4×3 $[1 \times 2]$	1×3 1×3 1×3 1×3 $[1 \times 2]$	3×3 3×3 $[2 \times 1]$ 1×3 1×3 $[1 \times 2]$	11×9 $[1 \times 4]$	9×9 $[1 \times 3]$
256 channels	3×3 3×3 $[2 \times 1]$	3×3 3×3	9×3 3×3	3×3 $[2 \times 1]$ 3×3 3×3 $[2 \times 1]$	4×3 3×3 $[2 \times 1]$ 3×3 3×3 $[2 \times 1]$	1×3 1×3 1×3 1×3	5×6 $[3 \times 3]$	3×4
output channel size	1×3							1×8

descent based back propagation algorithm.

Several specific FBANK configurations are used in the experiments. Note that Δ and $\Delta\Delta$ features are not utilized in the proposed very deep CNNs, since that the relatively wider context window size (mostly ≥ 17 in proposed very deep CNNs) is long enough to capture the dynamic information.

3.2. Feature extension

As described in [9], input channel size is 11×40 for hybrid CNN system, unfortunately, it's nearly impossible for such a small size input to survive through a very deep CNN with dimension-reducing operations.

Hence, model A_1, A_2, and A_3 are designed to firstly examine appropriate context window size of input features. In Table 2, the notion *effective convolution operations* is introduced. Since convolution operations whose kernel size is 1 on a dimension can be seen as a linear transformation of input channels followed by non-linearity, convolution operations with a non-one kernel size on that dimension are named as effective convolution operations. Similarly pooling operations with a size 1 on that dimension are not counted either.

Table 2[2] shows, wider context window size enables more temporal effective convolution operations, and model A_1 yields a significant improvement over model A_3, but employing even wider context window (model A_2) gives only a little gain. Additionally, too wide context window is impractical in real application, context window size is fixed to 17 for rest of the experiments.

Table 2: *WER (%) as a function of context window size.*

model	context window size	# of effective conv. operations	WER
A_1	17	4	39.7
A_2	33	6	39.5
A_3	11	2	42.5

Next, spectral dimension extension is addressed. FBANK features, which preserve locality information of speech data, are commonly used in CNNs. The dimension of FBANK is easy to increase from normal 24 or 40 to 64 or even larger for very deep structures. In our deep model configurations, as the convolutional output channel size is fixed to 1×3, the input spectral dimension is extended with respect to the convolutional depth, i.e. 40-dim, 52-dim, and 64-dim for the 6-layer, 8-layer and 10-layer network respectively.

3.3. Convolutional depth exploration

After feature extension investigation, model $\mathrm{A}_1, \mathrm{B}, \mathrm{C}_1$ are then designed to explore the effects on convolutional depth, with 6, 8, and 10 convolutional layers, respectively. As shown in Table 3, there is a large improvement when increasing the layer number from the normal 2 to 6, and the steady gains are still obtained when from 6 to 10. The proposed very deep CNNs show significant performance gain over the shallow CNN.

Table 3: *WER (%) as a function of convolutional depth.*

model	# of conv. layers	WER
baseline CNN	2	42.2
A_1	6	39.7
B	8	39.6
C_1	10	39.3

3.4. Delayed convolution operations

As described in Table 1, the first five networks $(\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \mathrm{B}, \mathrm{C}_1)$ with different convolutional depths are designed to have one same property, that both effective convolution operations and pooling operations are performed as close as possible to the toppest layers, which is believed better than the otherwise. To prove this assumption, the model C_2 is designed to perform effective convolution operations and pooling operations on temporal dimension from the very beginning. As shown in the Table 4, it's clear that model C_1

[2] Note that different from Table 1, only effective convolution operations are counted here.

outperforms model C_2. This kind of operation design is named as delayed convolution operations.

Table 4: *WER (%) on delayed convolution operations.*

model	delayed conv.	WER
C_1	yes	39.3
C_2	no	40.8

3.5. Effects of depth

Finally, to confirm the performance gain is mainly due to larger convolutional depth and to exclude the effects of feature extension, model D is designed to have the same convolutional input and output configuration as model C_1, but has only 2 convolutional layers similar to baseline CNN.

As shown in Table 5, the longer context indeed gets an improvement when compared to baseline CNN, however the very deep model C_1 obtains further improvement (5% relative) over model D. This confirms that the convolutional depth is the most important factor in the proposed very deep CNNs.

After careful exploration of possible structures of very deep CNNs, model C_1 in this section is proposed as *very deep CNN* for experiments in Section 4.

Table 5: *WER (%) on effects of depth.*

model	# of conv. layers	WER
baseline CNN	2	42.2
C_1	10	39.3
D	2	41.3

4. Experiments with the very deep CNN

In this section, the proposed very deep CNN is evaluated against baseline CNN and baseline DNN on 15-hr Callhome English and 51-hr Switchboard English.

4.1. Evaluation on 15-hr Callhome English

Results on 15-hr Callhome English are shown in Table 6 to compare the proposed very deep CNN and baseline DNN & CNN[3]. It is shown that baseline CNN is better than baseline DNN, and the proposed very deep CNN obtains a 9% and 7% relative improvement over baseline DNN and CNN respectively, which proves the effectiveness of the proposed very deep CNN preliminarily.

Table 6: *WER (%) comparison on 15-hr Callhome English.*

model	WER
baseline DNN	43.0
baseline CNN	42.2
very deep CNN	39.3

The number of model parameters is listed in Table 7, the deepest CNN in this work even has less parameters than both baseline CNN and baseline DNN. Actually, for most complex CNNs, parameters are mainly from the first fully connected layer. Nevetheless under the configuration that the convolutional output channel size is fixed to 1×3, the parameters are massively reduced. This demonstrates the effectiveness of the proposed very deep CNN again.

Table 7: *Comparison of model size.*

model	# of conv. layers	# of parameters
baseline DNN	-	27.6M
baseline CNN	2	21.1M
A_1	6	19.2M
B	8	19.9M
C_1	10	20.5M

4.2. Evaluation on 51-hr Switchboard English

During our experiments it is found that training a very deep CNN without parallelization is very time consuming, so we decide to evaluate the proposed very deep CNN on a subset of 309-hr Switchboard English.

A 51 hours subset (including 810 randomly chosen speakers) from the whole 309-hr Switchboard English Corpus is made as the new training and development set. The Switchboard portion of `Hub5'00` set (referred to as `swb`) and the Fisher portion of `rt03` set (referred to as `fsh`) are used as test sets.

The GMM-HMM system is built with 3001 triphone states, and then all NNs in the experiments are trained using the same configuration as those in 15-hr Callhome evaluation. Table 8 shows the performance on both `swb` and `fsh`, the proposed very deep CNN offers a 8-12% relative improvement over baseline DNN and a 4-6% relative improvement over baseline CNN. The improvement showed on 15-hr Callhome English could transfer to a larger 51-hr Switchboard task, which further confirms that the proposed very deep CNN can provide significant improvement over the shallow CNN.

Table 8: *WER (%) comparison on 51-hr Switchboard English.*

model	`swb`	`fsh`
baseline DNN	24.4	29.3
baseline CNN	23.0	28.2
very deep CNN	21.6	26.9

5. Conclusions

In this work, the possibility to apply very deep CNNs on LVCSR tasks is explored. To be better applied on speech, the span size of speech feature is fixed, the feature dimension is extended with respect to the convolutional depth, and further a novel way to apply convolution operations is proposed. Different from CNNs in previous works which have 2 convolutional layers at most, a very deep CNN with 10 convolutional layers is proposed after careful investigations. Eventually the very deep CNN proposed in this work offers a 8-12% relative improvement over baseline DNN system and a 4-7% relative improvement over baseline CNN system. The proposed very deep CNN will be evaluated on 309-hr Switchboard English in the future.

[3] The configuration of baseline DNN is 6 × 2048 with sigmoidal units in this paper, and baseline CNN is described in Section 3.

6. References

[1] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Interspeech 2011*. International Speech Communication Association, August 2011.

[2] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, Nov 2012.

[3] G. 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 (receiving 2013 IEEE SPS Best Paper Award)*, vol. 20, no. 1, pp. 30–42, January 2012.

[4] Y. LeCun and Y. Bengio, "Convolutional Networks for Images, Speech, and Time-Series," 1995.

[5] 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.

[6] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 4277–4280.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 10, pp. 1533–1545, Oct 2014.

[9] T. Sainath, A.-R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 8614–8618.

[10] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A. rahman Mohamed, G. Dahl, and B. Ramabhadran, "Deep convolutional neural networks for large-scale speech tasks," *Neural Networks*, vol. 64, no. 0, pp. 39–48, 2015, special Issue on Deep Learning of Representations.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[13] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *CoRR*, vol. abs/1301.3557, 2013.

[14] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, dec 2011, iEEE Catalog No.: CFP11SRW-USB.

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