Gated Convolutional LSTM for Speech Commands Recognition *

Dong Wang[1][0000−0002−6992−7950]✉, Shaohe Lv[1], Xiaodong Wang[1], and Xinye Lin[1]

Science and Technology on Parallel and Distributed Laboratoratory, National University of Defense Technology, Changsha, China
{wangdong08,shaohelv,xdwang}@nudt.edu.cn
lxytcmn@gmail.com

Abstract. As the mobile device gaining increasing popularity, Acoustic Speech Recognition on it is becoming a leading application. Unfortunately, the limited battery and computational resources on a mobile device highly restrict the potential of Speech Recognition systems, most of which have to resort to a remote server for better performance. To improve the performance of local Speech Recognition, we propose *C-1-G-2-Blstm*. This model shares Convolutional Neural Network's ability of learning local feature and Recurrent Neural Network's ability of learning sequence data's long dependence. Furthermore, by adopting the Gated Convolutional Neural Network instead of a traditional CNN, we manage to greatly improve the models capacity. Our tests demonstrate that *C-1-G-2-Blstm* can achieve a high accuracy at 90.6% on the Google Speech Commands data set, which is 6.4% higher than the state-of-art methods.

Keywords: Acoustic Speech Recognition · localize · Gated Convolutional Neural Network · Long Short Time Memory

1 Introduction

With the fast advancement of intelligent devices such as robots and smart phones, Acoustic Speech Recognition is becoming more and more popular in human-machine interaction. Speech assistants such as Google Now, Apple Siri, Microsoft Cortana are widely used around the world. To recognize the human speech accurately, most of these systems use a Client-Server (C/S) structure, where the speech recognition models with complex structure and high computing cost are put on cloud servers. The speech data is usually first collected on the mobile devices, then sent to the remote server. After the speech is processed and recognized, the result is then sent back to the mobile devices. Systems using this C/S can achieve good performance, but faces the following limitations. First,

* This work was supported by National Natural Science Foundation of China No.61472434, Science and Technology on Parallel and Distributed Laboratoratory Foundation No.9140C810109150C81002, National University of Defense Technology.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53

they depend heavily on a stable Internet connection, without which the system can't work. Second, sending data to a remote server introduces risks of privacy leak[3]. To circumvent these limitations, a speech recognition system that works completely locally is much in demand.

In practice, a few local speech recognition systems are already in deployment. For example, "Google Now" uses a local system to recognize a simple "OK Google" comamnd to wake up the main service. However, due to the limitation of hardware resources, mobile devices can only run relatively simple models, which have limited recognition capabilities. Therefore, most current local speech recognition systems only serve as a wake-up watchdog for more powerful online speech recognition services.

In order to improve the accuracy of local speech recognition systems, we design a deep neural network model based on Gated CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network). The model combines CNN's ability of learning local features and RNN's ability of learning the long-distance dependence features of sequential data. Our experiements show that the model achieves a high accuracy of 90.6% on the Google Speech commands dataset, outperforming the state-of-art work by 6.2%.

Our contribution is two-fold.

- First, we design an efficient model wich combines CNN and RNN. Compared with the existing CNN + RNN work[2, 23], our model uses fewer layers and a simpler neural network structure while achieving much higher recognition accuracy.
- Our model adopts the Gated CNN network structure. Compared with conventional CNN, Gated CNN uses self-attention-like operations and more nonlinear transformations, which effectively enhance the model's ability of selecting important features.

2 Related Work

CNN is originally designed for image identification, classification, etc. Since LeCun sucessfully trained a multi-layer net using CNN in LeNet[10], Deep Nerual Nets based on CNN achieve great success in image related tasks[6, 9, 15, 16]. By adopting the local receptive field, weight sharing, sub-sampling and other technologies, CNN is very robust with the translation and transformation in the data. It also has a strong ability to learn data's local patterns. These features make CNN a great tool in speech processing and natural language processing as well.

As a structure for handling time-sequence data, RNN focuses too much on the last input signals, and suffers from gradient explosion and the vanishing problem. As a result, RNN usually does not work well at its early stage. The proposal of Long-Short Time Memory (LSTM)[8] provides a good solution for these problems and greatly improves RNN's ability of learning long distance dependence. Benefited from LSTM, Gated Recurrent Unit (GRU)[4] and other

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53

structures, RNN has achieved a great breakthrough in natural language processing, translation and speech recognition.

In recent years, lots of works are using neural net to fulfill the speech processing task. Google Now[3] uses a fully connected Deep Neural Network (DNN) model to recognize the wake-up command "ok google". Compared with the Key-Word/Filter Hidden Markov Model, which is commonly used in existing Keyword Spotting system, this DNN model achieves 39% performance improvement. However, fully connected DNN ignores the structural patterns of the input data. No matter in what order is the input data organized, the fully connected DNN model will reach the same performance in the end, This will cause problems for speech recognition as the context of speech heavily relies on the speech data's structual feature in both the time and frequency domain. Besides, fully connected DNN methods cannot handle the translation invariance in the data. Different speakers or speaking styles can cause the formats translating in frequency domain, hence can hardly be processed with a fully connected DNN. Although theoretically full connected DNN can be trained with translation invariance, it requires lots of training data[14].

CNN's success in image domain demonstrates its ability to fix the disadvantages of fully connected DNN's. Inspired by this, CNN is more and more used in speech recognition[1, 5, 7, 22, 19]. [14] designs a CNN-based neural model, which achieves better recognition results than [3] while reducing the model's scale. [11] uses the CNN neural model and transfer learning, combined with Dilated Kernels for Multi-scale Inputs[21] to recognize speech commands. It builds a 121-layer neural net, pre-trains it on the UrbanSound8K dataset and achieves an accuracy of 84.35% on the Google Speech Commands dataset[20]. On the same data set, [17] designs a 15-layer deep residual net[6] combined with Dilated Kernels for Multi-scale Inputs. In the task of recognizing 12 commands selected from all 30 in Google Speech Commands, the model achieves an accuracy of 95.8%.

Although CNN outperforms fully connected DNN greatly in terms of recognition performance, it also has some disadvantages. The features learned by CNN are just local. Its scope is limited by the filter's shape and CNN's layer number, hence the features cannot cover the entire speech on either the time or frequency domain. To make the featurse cover a larger scope, the model has to be much deeper. Relatively, RNN shows much better performance in learning long distance dependence of sequential data. [2] proposes a CRNN neural model with 32 CNN layers and 1 RNN layer. Combining CNN's ability of learning local features and RNN's ability of learning long distance dependence, the CRNN neural model achieves an accuracy of 97.7% for detecting the occurrence of "TalkType". To solve the task of translating speech to text, [23] designs a 15-layer neural model based on ConvLSTM (one kind of LSTM which merges CNN inside) and CNN, in combination with many other technologies such as network-in-network, batch normalization, and residual connection. The model achieves a word error rate of 10.5% on the WSJ ASR dataset. However, all these models combining CNN and RNN have the problem of being too deep and complex.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53

Based on these work and aimed at improving the accuracy of local speech commands recognition, we propose a Gated Convolutional Recurrent Neural Network model. This model combines the advantages of Gated CNN and RNN networks and ultimately achieves a recognition accuracy of 90.6% on the Google Speech Commands dataset.

3 Model Design

Fig. 1 shows the basic structure of the proposed model, referred to as *C-1-G-2-Blstm*. The main part of the model is a multi-layer Gated CNN network connected with a bi-directional RNN network.

The model outputs the probabilities of the input speech being each command. For each speech, the input data firstly passes through a conventional two-dimensional convolution to increase feature maps, and then passes through multi-layer Gated CNN to extract local features on different scope scales. After local feature extraction, the model uses a bi-directional RNN to learn long distance dependence and obtains feature vector of the input speech. Finally, according to the feature vector, the model gives the prediction.

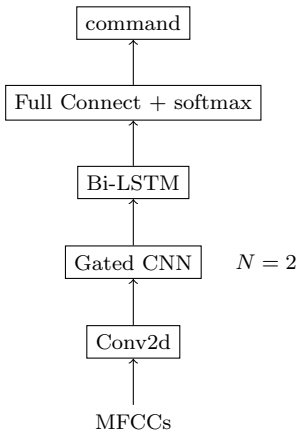

Fig. 1. Model Structure

3.1 Speech Preprocessing

For every speech fragment, its corresponding input to the model is the Mel-Frequency Cepstrum Coefficient (MFCC). In our test, we extract the MFCC with a frame window of 128 milliseconds, a frame offset of 31.25 milliseconds, and 20 filters. Finally, for each speech fragment, we get t MFCC frames with $f = 20$ dimensions, where t depends on the speech's duration.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53

3.2 CNN Net

CNN has shown a huge advantage in learning the data's local feature. Its successful application in image field inspired us that CNN can also be used to learn speech's time-local feature.

A nice feature of CNN is: as the number of CNN layer increases, the upper layer can have a larger receptive field, thereby extracting the feature of a larger local scope. Therefore, multi-layer CNN networks can help to analyze the speech's features at different scales.

The input of the CNN part is the MFCC frames x, where $x \in \mathbb{R}^{1\times t\times f}$. Most of the existing work treat MFCC as a two-dimension feature map with shape $t \times f$. Instead, in this paper we treat MFCC as f feature maps with dimension of $1 \times t$. This better coresponds to MFCC's physical meaning: different frequencies are different features. $F \in \mathbb{R}^{m\times n\times h\times r}$ denotes the kernel of two-dimensional convolution, where m and n denote the kernel's height and width, h denotes the number of input feature maps, and r denotes the number of output feature maps.

Conv2d. The first layer in our CNN part is a conventional 2-dimension convolution. In this layer the parameters are: $m = 1, h = t$ and $r > f$. Using these parameters has the following effects. First, with a $m \times n$ convolution in time domain, this layer can learn feature in local time. Second, $h = f$ makes different frequencies treated as different feature map. Third, by set $r > f$, this layer can recombine frequencies and produce more feature maps.

Gated CNN. The second and third CNN layer use Gated Convolution to further learn the local feature of the speech.

Gated convolutional layer is proposed in [12], its structure is shown in Fig. 2. Equtation (1) gives the definition of Gated Convolution, which is inspired by the multiplication gate in LSTM.

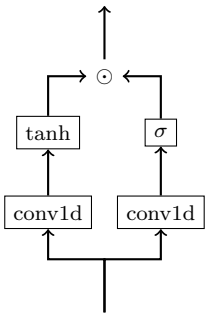

Fig. 2. Gated CNN

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53

$$y = \tanh(F_f * x) \odot \sigma(F_g * x) \tag{1}$$

In (1), $*$ is the convolution operation, σ is the sigmoid operation, $\odot$ denotes multiplication between corresponding elements, and F_f, F_g are the convolution kernels of two convolutions respectively.

Compared with conventional CNN, Gated Convolution introduces more non-linear operations and multiplication, which can improve the model's learning and expressing capacity. In addition, Self-Attention [18] is also obtained by multiplying the corresponding elements of tanh and σ.

3.3 RNN Net

CNN network can learn local features in different time periods. However, as time-series signal, speech's characteristics and contents are heavily related to its time order. The same local features appear at different time may have different meanings. This time-related feature can not be learned through CNN or full connected layer.

The successful application of RNN in natural language processing demonstrates its advantages in learning sequence features and long-range dependencies. Some work[1, 5] have recently applied RNN in speech recognition with a large vocabulary. In order to characterize the timing feature of the speech, we connect an Bi-directional LSTM network after CNN net. Fig. 3 shows the RNN network diagram.

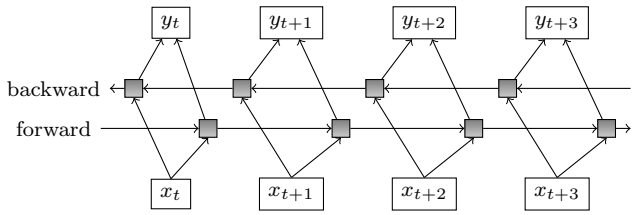

Fig. 3. RNN structure

For the RNN model, the critical point is how to establish the link between the previous information and the current state. As a classic RNN structure, LSTM performs the following steps on the input data. First, calculate the forgotten gate (2), the input gate (3), and the input information (4), second, update the hidden state (5), then the output gate (6), and finally calculate the current step's output according to the output gate and the hidden state (7).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53

$$\widetilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{4}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \tag{5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t \odot \tanh(C_t) \tag{7}$$

4 Experiments and Analysis

4.1 Dataset

In this paper we use the Google Speech Commands dataset. This dataset was released by Google in August 2017. It includes 65,000 speech data, covering thousands of people reading 30 commands, as well as some background noises. Most of these speech audios are mono, and last for a second, with a sampling rate of 16KHz, sampling resolution of 16bit. The division of training, validation and test set is shown in Tab. 1.

Table 1. Statistics of Google Speech Commands

Set	Train	Valid	Test
Scale	51,088	6,798	6,835

4.2 Experiment settings

To analyze the model from different aspects such as CNN network structure, network depth, the combination of CNN and RNN, and compare it with existing work, we design a variety of models with different structures and conduct extensive experiments. These models are as follows.

- C-p-G-q-Blstm/FullConnect: The model consists of p conventional 2-dimension CNN, q Gated CNN and a bidirectional LSTM (or fully connected layer). By adjusting the values of p, q, and choosing Blstm or FullConnect, we build a variety of different models for speech commands recognition.
- *Transfer Learning Network* [11]: This model pre-trains a 121-layer net on the UrbanSound8K dataset, and then transfers to recognize Google Speech Commands dataset.

In our experiments, each model is trained for specified epochs (it is found that most models can converge to their best performance in 100 epochs) on the training set, then select the best-performing model for evaluation. In order to accurately evaluate the model's performance and eliminate the influence of random factors, the experiment of each model is repeated 10 times. The average of these 10 results is taken as the final evaluation criterion.

For the model *Transfer Learning Network*, we use the result in [11] instead of reproducing it ourselves.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53

4.3 Experiment Results

Impact of Gated CNN's Depth. To explore the impact of Gated CNN's depth on speech recognition results, by using different number of Gated CNN layers(which means setting different value for q) in model *C-p-G-q-Blstm*, we get model *C-1-G-2-Blstm*, *C-1-G-5-Blstm*, *C-1-G-7-Blstm*, *C-1-G-9-Blstm*, *C-1-G-10-Blstm*, *C-1-G-20-Blstm*, *C-1-G-50-Blstm*. Table 2 gives the final recognition accuracy of these models.

Table 2. The impact of Gated CNN's Depth

model	Valid Accuracy(%)	Test Accuracy(%)
C-1-G-2-Blstm	**90.9**	**90.6**
C-1-G-5-Blstm	90.4	90.0
C-1-G-7-Blstm	89.7	89.5
C-1-G-9-Blstm	88.7	88.2
C-1-G-10-Blstm	88.2	87.9
C-1-G-20-Blstm	diverge	diverge
C-1-G-50-Blstm	diverge	diverge

Valid Accuracy and Test Accuracy represent the best model's recognition accuracy on the validation set and test set. Experiment results in Table 2 show that, for the Google Speech Commands dataset, deeper Gated CNN network does not necessarily have a better recognition performance. We can see that as the number of Gated CNN layer increases, the model's recognition performance firstly increases and then decreases, and when it reaches a certern number, the model does not converge.

This phenomenon may be caused by the limited amount of the data. A net with too many layers is too large and have too many parameters, which make it difficult to train the net effectively, so it can not achieve good results, or even fails to converge.

Experiment results show that the model *C-1-G-2-Blstm* with 2-layer Gated CNN achieves the best performance. In the follow-up experiments, this paper will use the model *C-1-G-2-Blstm* as the evaluation benchmark.

Impact of Gated Convolution. To analyze Gated CNN's help for speech commands recognition, we replace the Gated CNN in model *C-1-G-2-Blstm*, *C-1-G-5-Blstm*, *C-1-G-7-Blstm* with conventional CNN, getting models *C-3-G-0-Blstm*, *C-6-G-0-Blstm*, *C-8-G-0-Blstm*. Table 3 gives the comparison between the results of models before and after the replacement. From the results we can conclude that compared with the conventional CNN, Gated CNN can efficiently improve the model's prediction accuracy.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53

Table 3. the impact of Gated CNN

model	Valid Accuracy(%)	Test Accuracy(%)
C-1-G-2-Blstm	**90.9**	**90.6**
C-3-G-0-Blstm	87.2	87.2
C-1-G-5-Blstm	**90.4**	**90.0**
C-6-G-0-Blstm	86.9	86.7
C-1-G-7-Blstm	**89.7**	**89.5**
C-8-G-0-Blstm	83.5	83.2

Impact of CNN and RNN. To evaluate whether the combination of CNN and RNN could perform better than just CNN or just RNN, based on the model *C-1-G-2-Blstm*, we design another two models:

- *C-0-G-0-Blstm*: delete the CNN structure in *C-1-G-2-Blstm*, just keep the RNN structure.
- *C-1-G-2-FullConnect*: keep the CNN structure in *C-1-G-2-Blstm*, but replace the RNN structure with full connected layer.

Table 4 gives these models' experiment results. From the table we can see that, compared with the model *C-1-G-2-Blstm* which combines CNN and RNN, just using CNN or RNN results in a drastical decrease in recognition accuracy. Therefore, we can get the conclusion that combining the advantage of CNN and RNN is greatly helpful for speech command recognition.

Table 4. Comparison of CNN and RNNs Impact

model	Valid Accuracy(%)	Test Accuracy(%)
C-1-G-2-Blstm	**90.9**	**90.6**
C-0-G-0-Blstm	62.5	61.6
C-1-G-2-FullConnect	81.3	81.1

Comparison with Existing Works. In this paper we design two experiments to compare with Transfer Learning Network [11], which is the state-of-art work. Firstly, we compare the *C-1-G-2-Blstm* and Transfer Learning Network's recognition accuracy on all 30 commands in Google Speech Commands dataset. Results are shown in the second column of Table 5. Secondly, we re-train a new *C-1-G-2-Blstm* on the 20 commands selected in [11], and compare it with Transfer Learning Network. Results are shown in the third column of Table 5. From the results we can see that *C-1-G-2-Blstm* greatly outperforms Transfer Learning Network, both on all 30 commands and on the selected 20 commands.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53

Table 5. Comparison between *C-1-G-2-Blstm* and Transfer Learning Network

model	Test Accuracy_30(%)	Test Accuracy_20(%)
C-1-G-2-Blstm	**90.6**	**90.6**
Transfer Learning	84.4	82.1

Recognition Performance on Every Single Command. Table 6 gives *C-1-G-2-Blstm*'s recognition accuracy on every command. The accuracy decreases from left up to right down in turn. The command "happy" has the highest recognition accuracy of 97.2%, while the command "no" has the lowest recognition accuracy of 84.1%.

Table 6. Recognition Accuracy of Every Command

Comm[a]	Acc[b](%)	Comm	Acc(%)	Comm	Acc(%)
happy	**97.2**	five	92.6	bed	90.3
sheila	96.2	two	92.4	one	90.3
six	94.7	off	92.3	wow	90.2
house	94.7	marvin	91.4	dog	89.4
nine	94.2	up	91.2	bird	89.0
seven	94.1	stop	91.1	three	88.0
cat	94.0	right	91.1	go	86.9
eight	93.8	four	90.9	tree	85.5
left	93.6	on	90.7	down	85.3
yes	93.4	zero	90.4	no	84.1

[a]is abbreviation for "command".
[b]is abbreviation for "accuracy".

After analyzing all the 30 commands we can find that the command "happy" is special and different from other commands in pronunciation, so it's recognition accuracy is the highest. There are 7 commands whose recognition accuracy is below 90%: "dog", "bird", "three", "go", "tree", "down", "no". These commands are easy to be confused with the others. For example, "bird" is similar with "bed" in pronunciation. Main faults during recognizing these seven commands are given in Table 7.

For these 7 commands, we select everyone's most likely wrong recognition and fault probability. The first row in table 7 gives the groundtruth label, the first column gives the model's recognized label, the values represent the probability.

Take the second column as an example. It shows the distribution of fault recognition for command "no". From this column we can see that when mistakingly recognized, "no" is mistaken for "go" with a probability of 40%, and mistaken for "down" with a probability of 20%. In fact, "no", "go" and "down" do have similarities in pronunciation.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53

Table 7. tab:Main Faults in Recognition

	no	go	down	tree	three	bird	dog
no	-	**39.4**	**45.9**	-	-	-	21.1
go	**40.0**	-	**24.3**	-	3.1	5.9	**15.8**
down	**20.0**	**15.2**	-	-	-	5.9	**36.8**
tree	-	-	-	-	**53.1**	-	-
eight	-	3.0	-	7.1	**12.5**	-	-
right	-	3.0	-	-	-	**35.3**	-
bed	5.0	-	5.4-	-	-	**11.8**	-
three	-	-	2.7	**78.6**	-	-	5.3
two	-	12.1	2.7	**14.3**	6.2	-	-

From table 7 we can conclude that for those commands with low recognition accuracy, it's mainly because they are similar with some other commands in pronunciation, making it more difficult to distinguish them. This phenomenon shows us a new direction for future work: developing methods to distinguish similar speech commands.

4.4 Model FootPrint

Most models that combine CNN and RNN have a problem of being too deep and complex. For example, [2] proposes a CRNN neural model with 32 CNN layers and 1 RNN layer, [23] designs a 15-layer neural model that contains 8 CNN layers and 7 ConvLSTM (one kind of LSTM which merges CNN inside) layers. Comparing with these work, our *C-1-G-2-Blstm* only uses 3 CNN layers and 1 LSTM layers, greatly reducing the model's complexity. Parameters and multiplications used for the *C-1-G-2-Blstm* is shown in table 8.

Table 8. Parameters and multiplications used for the *C-1-G-2-Blstm*

layer	m	n	h	r	Par.	Mult.
Conv2d	1	5	20	64	6.25K	200K
Gated-Conv2d	1	5	64	64	40K	1282K
Gated-Conv2d	1	5	64	64	40K	1282K
Bi-LSTM	1	64	-	-	64.5K	2060K
FC	128	30	-	-	3.78K	3.75K

In our experiments, every training epoch takes 15 seconds, while every testing epoch takes 0.9 seconds. Considering that the test set contains 6,835 samples, *C-1-G-2-Blstm* can recognize about 7,000 commands per second.

Based on our *C-1-G-2-Blstm* we build an apk for android cellphones. To build the apk file, we first use TensorFlow's tool to freeze our computing graph into a

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53

pb file, which is only 911kb. Then we build an android apk which use the frozen graph to perform speech commands recognition, the apk is only 22M.

5 Summary and Future Works

For the task of speech command recognition on mobile device, this paper designs a model *C-1-G-2-Blstm* based on Gated CNN and bidirectional LSTM. This model uses CNN to learn the speech's local features, RNN to learn sequence long-distance dependence features, and Gated CNN to improve the model' capacity. Compared with existing work based on CNN and RNN, our model uses fewer layers and simpler net structure. Finally *C-1-G-2-Blstm* achieves an accuracy of 90.6% on the Google Speech Commands dataset, outperforming the existing state-of-art work by 6.4%.

One of our future work is to further improve the model's recognition performance. [13] points out that the preprocessing methods of speech data, the usage of batch normalization and other technologies such as dilated convolution will affect the model's performance. We are going to conduct experiments on more datasets to evaluate these factors' impact. On the other hand, because speech recognition especially wakeup-word recognition is seriously limited by local hardware resources, it is also a very important development direction to explore how to minimize the model size and computational complexity while ensuring the recognition accuracy.

Acknowledgment

This work is supported by the National Natural Science Foundation of China No.61472434, Science and Technology on Parallel and Distributed Laboratory Foundation No.9140C810109150C81002.

References

1. Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al.: Deep speech 2: End-to-end speech recognition in english and mandarin. In: International Conference on Machine Learning. pp. 173–182 (2016)
2. Arik, S.O., Kliegl, M., Child, R., Hestness, J., Gibiansky, A., Fougner, C., Prenger, R., Coates, A.: Convolutional recurrent neural networks for small-footprint keyword spotting. arXiv preprint arXiv:1703.05390 (2017)
3. Chen, G., Parada, C., Heigold, G.: Small-footprint keyword spotting using deep neural networks. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. pp. 4087–4091. IEEE (2014)
4. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53

5. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al.: Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567 (2014)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
7. Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., et al.: Cnn architectures for large-scale audio classification. In: Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. pp. 131–135. IEEE (2017)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
11. McMahan, B., Rao, D.: Listening to the world improves speech command recognition. arXiv preprint arXiv:1710.08377 (2017)
12. van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. In: Advances in Neural Information Processing Systems. pp. 4790–4798 (2016)
13. Sainath, T.N., Kingsbury, B., Mohamed, A.r., Dahl, G.E., Saon, G., Soltau, H., Beran, T., Aravkin, A.Y., Ramabhadran, B.: Improvements to deep convolutional neural networks for lvcsr. In: Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. pp. 315–320. IEEE (2013)
14. Sainath, T.N., Parada, C.: Convolutional neural networks for small-footprint keyword spotting. In: Sixteenth Annual Conference of the International Speech Communication Association (2015)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
16. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
17. Tang, R., Lin, J.: Deep residual learning for small-footprint keyword spotting. arXiv preprint arXiv:1710.10361 (2017)
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need. ArXiv e-prints (Jun 2017)
19. Wang, Y., Getreuer, P., Hughes, T., Lyon, R.F., Saurous, R.A.: Trainable frontend for robust and far-field keyword spotting. In: Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. pp. 5670–5674. IEEE (2017)
20. Warden, P.: Launching the speech commands dataset. Google Research Blog (2017)
21. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
22. Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Bengio, C.L.Y., Courville, A.: Towards end-to-end speech recognition with deep convolutional neural networks. arXiv preprint arXiv:1701.02720 (2017)
23. Zhang, Y., Chan, W., Jaitly, N.: Very deep convolutional networks for end-to-end speech recognition. In: Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. pp. 4845–4849. IEEE (2017)

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_53