Vol.2018-MUS-118 No.1
Vol.2018-SLP-120 No.1
2018/2/20

Distilling Knowledge from a Multi-scale Deep CNN Ensemble for Robust and Light-weight Acoustic Modeling

Michael Heck[1], Masayuki Suzuki[2], Takashi Fukuda[2], Gakuto Kurata[2], Satoshi Nakamura[1]

[1] Nara Institute of Science and Technology
[2] IBM Research AI

Abstract: This paper presents our work on constructing a multi-scale deep convolutional neural network (CNN) ensemble for robust and light-weight acoustic modeling. Several VGG nets are used that differ solely in the kernel size of the convolutional layers. The ensemble serves as teacher for distilling knowledge into a much simpler student CNN. We compare the performance of the distilled CNN model with the results of system combination. We show that the knowledge distillation from a multi-scale ensemble yields equal performance with the best conventional combination methods, with a much simpler system architecture and decoding pipeline.

1 Introduction

Building large-scale systems for automatic speech recognition (ASR) is a complex task. Model combination is a popular method to improve outputs of very large high performance systems [1]. However, finding good pairs for combination is not trivial.

We describe our approach to building a large but lightweight ASR system that can be used for offline transcription of massive data amounts. We train a set of deep convolutional neural network models with VGG net architecture [2] for combination that is inspired by the multi-scale convolutional neural network (MS-CNN) [3]. The MS-CNN consists of subnetworks with receptive fields that vary in size. The idea is that these receptive fields match objects of different scales, which help with robust recognition. We compare the model combination methods ROVER and posterior prediction fusion to the model compression method. Lastly, we perform knowledge distillation [4] from ensembles [5], where the knowledge of a "teacher" VGG ensemble is compressed – or distilled – into a "student" CNN model [6]. A CNN trained in this way is much simpler in structure but yields equal performance.

2 Multi-scale ensemble

Our *multi-scale ensemble* is a combination of VGG nets that are trained on the same data but differ in the size of their receptive fields. In VGG nets, the large convolutional kernels of standard CNNs are replaced with small kernels that are arranged in stacks of layers. This change produces the same receptive field with less parameters.

We use the "WDX" network layout of [7]. A graphical representation is given in Figure 1. The default network uses a kernel size of 3x3 (frequency domain x time domain). To keep the output size of the feature extraction sub-network before the fully connected layers the same for each model, we adjust the zero padding and max pooling accordingly. The networks take stacked logMel feature vectors as input. The last fully connected layer of the network represents context dependent (CD) HMM states. The output is a vector of posterior probabilities.

2.1 Models

To produce candidates for model ensemble, we copy the base model architecture and modify the kernel size in the frequency domain X and time domain Y. We allow variants with any size in {2,3,4}x{2,3,4} and also use 5x5, 6x6 and 7x7 kernels (see Figure 1). We adjust the zero padding for the convolutional layers and the max pooling size to compensate for changes in the kernel size.

2.2 Conventional combination

We build model ensembles by means of late model combination. We use the AMs from above to decode

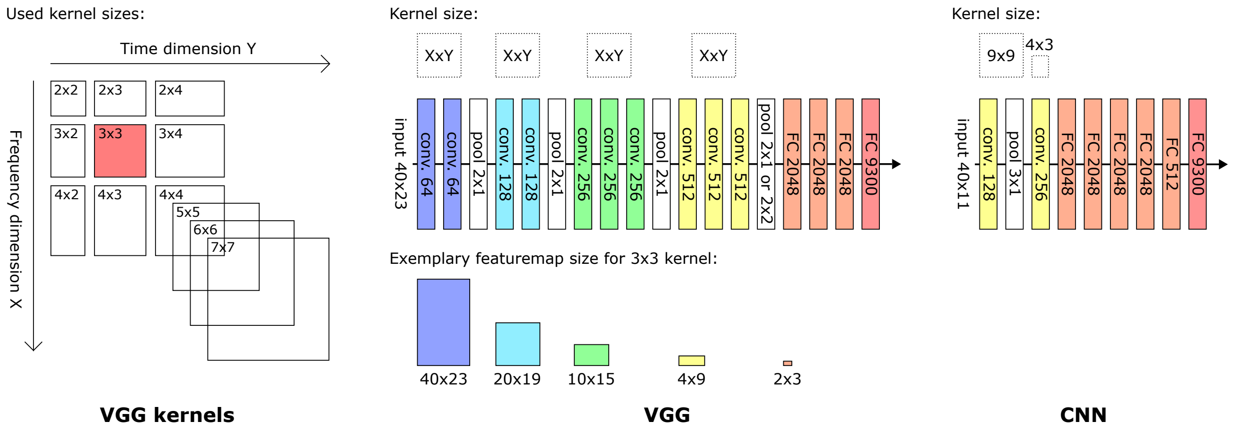

Figure 1: *Left:* Permitted VGG kernel sizes. *Center:* Teacher VGG net. *Right:* Student CNN.

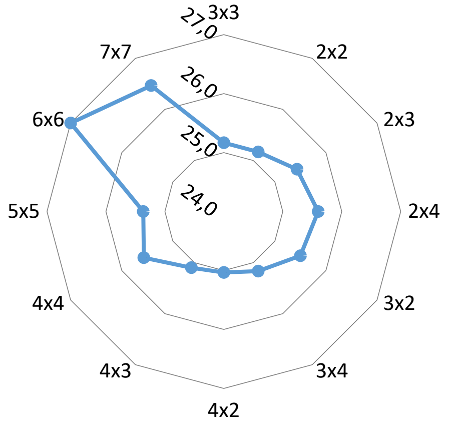

Figure 2: Performance of single VGG nets in WER.

the target data multiple times for ROVER combination using without considering confidence measures. Alternatively, we combine models by posterior probability fusion. For each feature vector $\vec{x}_i$ of frame i, we compute the non-weighted average of posterior probabilities for all CD states $\vec{s}_i$.This is straightforward as all M AMs have the same layout. After posterior probability combination, we complete the decoding. This has the advantage that full decoding has do be done only once.

3 Distillation from ensembles

Knowledge distillation is a method for model compression and knowledge transfer that takes a teacher model to train or guide a student model. The guidance is given by soft outputs (posterior probabilities). Knowledge is *distilled* by increasing the temperature T in the tempered softmax function in the last layer of a neural network to convert logits into posterior probabilities.

A temperature higher than the default $T = 1$ produces a softer probability distribution over the CD states of the network. Typically, the distilled model shows competitive performance with the advantage of being less complex. Ensembles of models can also serve as teacher. We use an ensemble of multi-scale VGG nets as teacher to train a classical CNN student with much simpler architecture (see Figure 1).

We construct the teacher ensemble by averaging the CD state posterior probabilities of the single VGG nets. We use the distillation for knowledge transfer pre-training. The student model is trained with soft labels from the ensemble as targets to push the model parameters into a good direction. What follows is a fine-tuning with the original hard labels to reach a good local optimum.

4 Experiments

Our data is conversational interview-style English (50h for training, 8.6h for testing).We use the cross-entropy criterion for training. The VGG nets are implemented in torch [8]. The initial learning rate is set to 0.03 and is divided by 3 after 25M, 30M and 35M frames. Training is stopped after 40M frames. The CNN is built with an in-house toolkit. The initial learning rate is 0.005 for knowledge distillation pre-training and 0.0005 for fine-tuning and is reduced with Newbob.

4.1 Multi-scale VGGs

Figure 2 shows that rhe VGG nets with kernel sizes circling around the default all performed similarly well. The default achieved 25.2% WER. The poorest model is using the 6x6 kernel and yields 27.0%

Table 1: Combination performance in WER.

#VGGs	1	2	3	6	9
ROVER	25.2	-	24.8	24.3	**24.0**
Posterior	25.2	24.5	24.3	24.1	**24.0**

WER. With larger kernels, the training duration is also growing considerably larger.

4.2 ROVER combination

ROVER proved very effective, even though most models achieve similar WER. The more models there are in the combination, the better the results. If we use squared kernels only, combining all systems from kernel size 2x2 to 7x7 produced the best hypotheses. The best ROVER combination was achieved by using the 9 systems whose kernel sizes circle around the 3x3 default (see Table 1).

4.3 Posterior combination

For posterior probability fusion, we extract and average the logit values for each frame from the layer right before the final softmax layer, for all networks of the ensemble. Our results in Table 1 show that the posterior combination is superior when only a few models are involved in the combinations: A 3-system posterior combined ensemble can compete even with a 6-system ROVER. We can also create 2-system ensembles, which is not possible with ROVER. Both methods converge to the same performance with increasing ensemble size.

4.4 Knowledge distillation

Our student CNN is already *experienced* in that it has been initialized by a training on about 2000h of out-of-domain Switchboard data. The soft labels from the ensemble are computed with the softmax function using a temperature $T = 2$.

The initialized CNN achieves 41.7% WER, a domain-adapted CNN by fine-tuning using the original hard labels 24.8% WER. The domain-adapted CNN can beat the single best VGG, but not our best multi-scale VGG ensemble. Knowledge distillation alone, i.e., the pre-training (pt) on soft labels, does not beat the domain-adapted model. However, fine-tuning (ft) with hard labels greatly increases performance so that the fine-tuned CNN model outperforms the domain-adapted model and achieves equal performance with the best VGG ensemble (see Table 2), but with much simpler structure.

Table 2: Distilled CNN performance in WER.

#VGGs	1	6	9
VGG → CNN (pt)	25.8	25.3	25.3
VGG → CNN (ft)	24.3	24.2	**24.1**

Table 3: Initial results on Switchboard 300h in WER.

Single			Combination	
2x2	3x3	4x4	ROVER	Posterior
12.4	12.3	12.0	11.8	11.8

4.5 Scalability

Table 3 lists results of scalability experiments on the Switchboard 300h data set. As can be seen, both ROVER and posteriorgram combination clearly improve decoding performance, even with just three multi-scale VGGs in the model ensemble. We also analyzed the impact of the kernel size on the real-time factor (RTF) for the model training with a single GPU (NVIDIA Tesla K80). Using the default 3x3 kernel yields an RTF of 0.12. The lowest (0.08) and highest (0.5) RTFs are achieved with the smallest (1x2) and largest (7x7) kernels, respectively.

5 Conclusions

We have shown that deep convolutional neural network acoustic models that solely differ in their kernel size are sufficiently diverse for model combination, which greatly simplifies building ensembles. Both ROVER and posterior combination of multi-scale models improve recognition accuracy. We successfully distilled the knowledge of ensembles into a classical CNN with much simpler architecture and achieved equal performance compared to the best performing conventional combination.

Acknowledgements

Part of this work was supported by JSPS KAKENHI Grant Number JP17H06101.

References

[1] George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, et al., "English conversational telephone speech recognition by humans and machines," *arXiv preprint arXiv:1703.02136*, 2017.

[2] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 354–370.

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[5] Yevgen Chebotar and Austin Waters, "Distilling knowledge from ensembles of neural networks for speech recognition," *Interspeech 2016*, pp. 3439–3443, 2016.

[6] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on*. IEEE, 2013, pp. 8614–8618.

[7] Tom Sercu, Christian Puhrsch, Brian Kingsbury, and Yann LeCun, "Very deep multilingual convolutional neural networks for LVCSR," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4955–4959.

[8] Ronan Collobert, Samy Bengio, and Johnny Mariéthoz, "Torch: a modular machine learning software library," Tech. Rep., Idiap, 2002.