Acoustic Model Compression with Knowledge Transfer*

Jianyan Yi1,2, Jianhua Tao1,2,3, Zhengqi Wen1, Ya Li1, Hao Ni1,2
1. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
2. School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China
3. CAS Center for Excellence in Brain Science and Intelligence Technology, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

Abstract: Mobile devices have limited computing power and limited memory. Thus, large deep neural network (DNN) based acoustic models are not well suited for application on mobile devices. In order to alleviate this problem, this paper proposes to compress acoustic models by using knowledge transfer. This approach forces a large teacher model to transfer generalized knowledge to a small student model. The student model is trained with a linear interpolation of hard probabilities and soft probabilities to learn generalized knowledge from the teacher model. The hard probabilities are generated from a Gaussian mixture model hidden Markov model (GMM-HMM) system. The soft probabilities are computed from a teacher model (DNN or RNN). Experiments on AMI corpus show that a small student model obtains 2.4% relative WER improvement over a large teacher model with almost 7.6 times compression ratio.

Keywords: model compression; knowledge transfer; deep neural networks; automatic speech recognition

1 Introduction

Deep neural networks (DNNs) have recently showed state-of-the-art performance in automatic speech recognition (ASR) tasks [1-6]. They have become the dominant acoustic modeling approach for large vocabulary continuous speech recognition. With a wide use of mobile devices, the industry has strong interests in utilizing DNN based models on devices, such as mobile phones and smart watches. However, these devices have limited computing power and limited memory. Unfortunately, top-performing systems usually use very deep and wide acoustic models with many parameters. One drawback of such large models is time consuming at decoding time. Another is that having large amounts of parameters results in high memory demanding. So large acoustic models are not well suited for application on mobile devices.

There are several attempts in the literatures to address this problem at decoding stage. These approaches can be roughly classified into the following categorizations: low rank matrix [7-10], frame skipping [11-12], vector quantization [13-14], Kullback-Leibler (KL) divergence [15-16], knowledge distillation [17-19] and hashing tricks [20]. The above mentioned approaches are able to achieve faster speed or less storage. However, few of them can achieve significant compression without any accuracy loss.

This paper proposes to use knowledge transfer to compress large acoustic models without performance loss for mobile devices. The concept of knowledge distillation has been around for a decade [17-18]. A more general framework is proposed by Hinton et al. [19] to transfer knowledge efficiently by using high temperature. At a high level, distillation involves training a new model, which is called a student model. The student model is trained to mimic the output distribution of a well-trained model which is called a teacher model. Inspired by the KLD-regularized model adaptation [21-23], we treat the output probability distribution of a large teacher model as a regularization term to generalize a small student model. The

---

*Grant Project: This work is supported by the National High-Tech Research and Development Program of China (863 Program) (No.2015AA016305), the National Natural Science Foundation of China (NSFC) (No.61425017, No.61403386), the Strategic Priority Research Program of the CAS (GrantXDB02080006).

Author: Jianyan Yi (1984-), female (Han), YUNNAN, PhD candidate.
Coresponding Contact: Jianhua Tao, Research fellow, jhtao@nlpr.ia.ac.cn
student model is trained with a linear interpolation of hard probabilities and soft probabilities to learn generalized knowledge from the teacher model. The hard probabilities are generated from a Gaussian mixture model hidden Markov model (GMM-HMM) system. The soft probabilities are computed from the teacher model using a forward pass. The experiments on RASC663 and AMI corpus show that the proposed method can achieve obvious model compression without accuracy loss. Moreover, results on AMI corpus demonstrate that a small model obtains about 2.4% relative WER improvement over a large model with almost 7.6 times compression ratio.

The rest of the paper is organized as follows. Section 2 discusses some related work briefly. Section 3 describes the proposed compression method. Section 4 introduces the framework of the proposed method. Section 5 presents the experiments. The results are discussed in Section 6. This paper is concluded in Section 7.

2 Related work

There are a few attempts to compress models using knowledge distillation. The work [15, 16] is to train a small DNN model by utilizing KL divergence to minimize the output of the two models. A small model is trained only with the soft probabilities. The method [19] is proposed to distill a single model from ensemble of models with high temperature. Most recently, the work [25] is to distill ensemble of models into a single model using KL divergence. In computer vision tasks, the work [24] is proposed to train deeper and thinner networks using hints training with output and intermediate information.

However, our work focuses on compressing a large acoustic model into a small acoustic model. Our approach is inspired by the KLD-regularized model adaptation [21]. The compression is performed not at a high temperature but at a temperature of 1. The student model is trained with an interpolation of hard probabilities and soft probabilities to learn generalized knowledge from the teacher model.

3 Proposed compression method

In this section, the proposed compression method is described in detail.

The KLD-regularized adaptation is first proposed in [21]. Inspired by the KLD-regularized adaptation, this paper proposes to compress acoustic models with knowledge transfer for mobile devices.

This paper treats the output probability distribution of a large teacher model as a regularization term to generalize a small student model. If the teacher model generalizes well, the student model will generalize in the same way. This student model will also obtain much better performance on the test set than the student model trained in the normal way on the same training set.

The regularization term is added to the standard cross entropy loss function $L_{\text{hard}}$. Given an input $x_i$ and an output label $y_{ij}$, the standard loss function $L_{\text{hard}}$ is formulated in Eq. (1).

$$L_{\text{hard}} = \sum_i \sum_j t_{ij} \log p(y_{ij} | x_i)$$  \hspace{1cm} (1)

Where $i$ is a sample id (frame id) in the training set and $j$ denotes an output label id (senone id), $t_{ij}$ is the hard (true) probability for the output label $y_{ij}$, $p(y_{ij} | x_i)$ is the posterior probability. The regularized loss function is depicted as follow:

$$L = (1 - \rho)L_{\text{hard}} + \rho \sum_i \sum_j q_{ij} \log p(y_{ij} | x_i)$$ \hspace{1cm} (2)

where $\rho$ is the interpolation weight, $q_{ij}$ is the soft (posterior) probability computed from the teacher model with forward pass. Eq. (2) can be reorganized to Eq. (3).

$$L = \sum_i \sum_j ((1 - \rho) t_{ij} + \rho q_{ij}) \log p(y_{ij} | x_i)$$  \hspace{1cm} (3)

where we define:

$$\bar{t}_{ij} = (1 - \rho) t_{ij} + \rho q_{ij}$$ \hspace{1cm} (4)

By comparing Eq. (1), Eq. (3) and Eq. (4), we can see that adding a regularization term to the
standard cross entropy loss function is equal to changing the target probability from the hard probability $t_{ij}$ to $\tilde{t}_{ij}$.

An excellent property of Eq. (5) is that the training of the student model can be simply performed with normal back propagation algorithm. The only thing that needs to be changed is the error signal at the output layer, which is now defined as a new probability $\tilde{t}_{ij}$. Eq. (5) shows that the student model is trained with the new probability to learn generalized knowledge from the teacher model.

The interpolation weight can be adjusted, typically using a development set. When interpolation weight is set to 1, the student model is trained only with soft probabilities. When interpolation weight is set to 0, the student model is trained only with hard probabilities.

4 Framework of the proposed method

The framework of the proposed method is shown in Fig. 1. All student models are smaller than teacher models.

If a student model is trained with labeled data, hard probabilities are produced from a GMM-HMM system. A large DNN or a recurrent neural network (RNN) based acoustic model is trained as a teacher model. Then soft probabilities are computed from the teacher model with a forward pass. A small DNN based student model is trained with new probabilities which are the linear interpolation of hard probabilities and soft probabilities. A hard probability is a one-hot vector, such as $[1 0 0 0]$. A soft probability has rank information for incorrect labels, like $[0.9 0.01 0.02 0.07]$. We can see that the soft probability can provide more information than the hard probability.

If a student model is trained with unlabeled data, soft probabilities are generated from an existing large DNN or RNN based teacher model. Then the student model is trained only with soft probabilities.

5 Experiments

To evaluate the proposed method, our experiments are conducted on two corpora: RASC863 [26] and AMI [27]. The compression is implemented based on Kaldi speech recognition toolkit [28].

The feature vector is 40-dimensional filter bank (FBANK) energies calculated on a 25ms window every 10ms. The context window of DNN is 11 frames. Bidirectional long short term memory (BLSTM) RNN based models use a single frame as input. The language model (LM) is a 3-gram LM. The used vocabulary has 80K words and the decoder is based on weighted finite-state transducers (WFST). The training terminates on the development set with a little improvement. The initial learning rate is set to $2 \times 10^{-3}$ and $8 \times 10^{-5}$ for DNN and BLSTM respectively.

In all experiments, $L$ denotes the number of hidden layers. $N$ denotes the number of hidden nodes. $Cmp.$ denotes the compression ratio for an acoustic model. $Stor.$ denotes the size of an acoustic model stored in hard disks. $Mem.$ denotes the size of an acoustic model loaded in memory. $RT100$ denotes real-time factor assuming 100 frames/second.

5.1 Mandarin corpus: RASC863

RASC863 is a Mandarin corpus which contains 4 regional accents, namely Chongqing, Shanghai, Guangzhou and Xiamen. The training set has 25612 utterances about 50 hours. The development set has 2561 utterances about 5 hours. The test set has 2676 utterances about 5.5 hours. There is no overlap among these data sets. The LM is trained on the transcriptions of the training set about 2.6M.

In order to develop methods to effectively compress large models, three group of experiments are conducted on this corpus. Firstly, a large DNN
model is trained with 4 hidden layers, 1024 hidden nodes and 4237 output nodes (senones). This model is used as a teacher model (T-DNN).

1) Interpolation weight for compression

The first group of experiments are conducted to explore the relationship between the interpolation weight and the performance of student models.

The large teacher model is T-DNN. The DNN based small student models are listed in Table 1, denoted as S1, S2, S3, S4 and S5 respectively. The interpolation weight \( \rho \) is set to 0.0, 0.2, 0.5, 0.8 and 1.0 for all student models. The weight is adjusted on the development set.

Table 2 presents WERs of the student models with different \( \rho \). We can see that when \( \rho \) is set to 0.0, all student models obtain the worst performance. When \( \rho \) is set to 0.8, all student models achieve the best performance. So \( \rho \) is set to 0.8 in the rest experiments.

2) Width and depth for student models

This group of experiments explore how the performance of student models are affected by their width and depth. The teacher model is T-DNN. There are four types of DNN based student models described in Table 3: thinner, shallower, shallower & thinner and deeper & thinner. The results of the experiments are listed in Table 3.

We can make some observations from Table 3. The model can be significantly compressed, when hidden nodes are reduced. The compression ratio is too low, when we only reduce hidden layers. We can obtain the highest compression ratio 11.2, when \( L = 3 \) and \( N = 128 \). However, the performance of this student model is very poor. When \( L = 5 \) and \( N = 256 \), the student model outperforms the teacher model with 4.8 times compression ratio and only occupies 33M memory.

3) Various teachers for a student model

This group of experiments are designed to explore the performance of a student model guided by different teacher models. Inspired by the above experiments, we train a DNN model with 5 hidden layers and 256 hidden nodes as the student model (S-DNN). There are two teacher models: T-DNN and T-BLSTM. T-DNN is same as the above experiments. T-BLSTM is a BLSTM based teacher model. The size of two teacher models is similar. T-BLSTM has 4 hidden layers, 560 cells. The output layer still has 4237 nodes. The results are listed in Table 4.

Table 4 shows that two students model both outperform their teacher models with 4.8 times compression and occupy less memory. The student model guided by T-BLSTM obtains the best performance and the WER is 35.05%. The student models also decode faster than the teacher model with 1.57~1.80 times speedup.

Table 1. Student models with different depth and width.

<table>
<thead>
<tr>
<th>Model</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>N</td>
<td>512</td>
<td>256</td>
<td>128</td>
<td>1024</td>
<td>256</td>
</tr>
</tbody>
</table>

Table 2. WERs (%) of five student models (S1-S5) with different interpolation weights on RASC863 test set.

<table>
<thead>
<tr>
<th>( \rho )</th>
<th>S1(%)</th>
<th>S2(%)</th>
<th>S3(%)</th>
<th>S4(%)</th>
<th>S5(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>39.72</td>
<td>42.00</td>
<td>46.52</td>
<td>39.74</td>
<td>41.16</td>
</tr>
<tr>
<td>0.2</td>
<td>38.56</td>
<td>40.29</td>
<td>44.36</td>
<td>38.42</td>
<td>39.83</td>
</tr>
<tr>
<td>0.5</td>
<td>37.53</td>
<td>39.15</td>
<td>43.08</td>
<td>37.53</td>
<td>38.56</td>
</tr>
<tr>
<td>0.8</td>
<td><strong>37.48</strong></td>
<td><strong>38.63</strong></td>
<td><strong>42.10</strong></td>
<td><strong>37.36</strong></td>
<td><strong>38.49</strong></td>
</tr>
<tr>
<td>1.0</td>
<td>38.02</td>
<td>39.08</td>
<td>38.52</td>
<td>38.03</td>
<td>38.91</td>
</tr>
</tbody>
</table>

Table 3. Results for different types of student models on RASC863 test set.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T-DNN</td>
<td>4</td>
<td>1024</td>
<td>38.62</td>
<td>-</td>
<td>46.0M</td>
<td>102M</td>
</tr>
<tr>
<td>S-thinner</td>
<td>4</td>
<td>512</td>
<td>37.48</td>
<td>2.3x</td>
<td>20.0M</td>
<td>64M</td>
</tr>
<tr>
<td>S-thinner shallower</td>
<td>4</td>
<td>128</td>
<td>42.10</td>
<td>10.2x</td>
<td>4.5M</td>
<td>19M</td>
</tr>
<tr>
<td>S-deeper thinner</td>
<td>3</td>
<td>1024</td>
<td><strong>37.36</strong></td>
<td>1.1x</td>
<td>42.0M</td>
<td>94M</td>
</tr>
<tr>
<td>S-deeper shallower</td>
<td>2</td>
<td>1024</td>
<td>37.99</td>
<td>1.2x</td>
<td>38.0M</td>
<td>86M</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1024</td>
<td>40.13</td>
<td>1.4x</td>
<td>34.0M</td>
<td>75M</td>
</tr>
<tr>
<td>S-thinner shallower</td>
<td>3</td>
<td>512</td>
<td>37.95</td>
<td>2.4x</td>
<td>19.0M</td>
<td>63M</td>
</tr>
<tr>
<td>S-deeper thinner &amp; shallower</td>
<td>3</td>
<td>256</td>
<td>38.92</td>
<td>5.1x</td>
<td>9.1M</td>
<td>32M</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>128</td>
<td>42.34</td>
<td><strong>11.2x</strong></td>
<td>4.1M</td>
<td>18M</td>
</tr>
</tbody>
</table>

Table 4. Results of a student model with different teacher models on RASC863 test set.

<table>
<thead>
<tr>
<th>Model</th>
<th>WER(%)</th>
<th>RT100</th>
<th>Stor.</th>
<th>Mem.</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-DNN</td>
<td>38.62</td>
<td>1.32</td>
<td>46.0M</td>
<td>102M</td>
</tr>
<tr>
<td>S-DNN</td>
<td>38.49</td>
<td>0.84</td>
<td>9.6M</td>
<td>33M</td>
</tr>
<tr>
<td>T-BLSTM</td>
<td>35.18</td>
<td>1.48</td>
<td>45.0M</td>
<td>111M</td>
</tr>
<tr>
<td>S-DNN</td>
<td><strong>35.05</strong></td>
<td>0.84</td>
<td>9.6M</td>
<td>33M</td>
</tr>
</tbody>
</table>
5.2 English corpus: AMI

The AMI is an English corpus which consists of 100 hours meeting recordings including close-talking and far-field microphones etc. We use the close-talking data which is collected from individual headset microphones (IHM). In our experiments, acoustic models are microphone independent. The training set has 108221 utterances about 82 hours. The development set has 13059 utterances about 9.5 hours. The test set has 12612 utterances about 8.5 hours. There is no overlap among these data sets. The LM is trained on the transcriptions of the training set about 7.9M.

This group of experiments empirically investigate the benefits of our method by comparing various student models trained only with hard probabilities (hard DNN), KL [16] or our proposed knowledge transfer based method (KT).

Firstly, a DNN model is trained with 6 hidden layers, 2048 hidden nodes and 2687 output nodes. This DNN model is used as the teacher model (T-DNN). Motivated by the above experiments, all student models are DNN based and have 4 hidden layers and 512 hidden nodes. The results of the student models are listed in Table 5.

<table>
<thead>
<tr>
<th>Model</th>
<th>WER(%)</th>
<th>Cmp.</th>
<th>Stor.</th>
<th>Mem.</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-DNN</td>
<td>35.24</td>
<td>-</td>
<td>144M</td>
<td>341M</td>
</tr>
<tr>
<td>hard DNN</td>
<td>37.17</td>
<td>7.6×</td>
<td>19M</td>
<td>61M</td>
</tr>
<tr>
<td>hard DNN-sMBR</td>
<td>36.73</td>
<td>7.6×</td>
<td>19M</td>
<td>61M</td>
</tr>
<tr>
<td>KL [16]</td>
<td>35.95</td>
<td>7.6×</td>
<td>19M</td>
<td>61M</td>
</tr>
<tr>
<td>proposed KT</td>
<td>34.42</td>
<td>7.6×</td>
<td>19M</td>
<td>61M</td>
</tr>
</tbody>
</table>

Table 5 shows that our proposed method KT achieves the best performance and the student model outperforms the teacher model. When the student models are only trained with hard probabilities, the performance of the student models will decrease. Although the performance of the method KL [16] is improved compared with hard DNN model, it still has some accuracy loss against the teacher model T-DNN.

6 Discussion

The above experiments show that our proposed method is effective. We make some interesting observations as follow.

When compressing models, reducing the width of the student model is more effective than reducing the depth.

Since the output layer has a large number of output labels and depth leads more abstract representations at a higher layer.

Increases in the accuracy of the teacher models yield similar increases in the accuracy of the student model. Since the teacher model has higher accuracy, the student model will correct more errors.

The performance will decrease, if the student model is trained only with hard probabilities or soft probabilities. The student model will obtain better performance, when the interpolation weight of soft probabilities is higher than hard probabilities.

Our proposed method on AMI corpus achieves the best performance and the student model outperforms the teacher model with 7.6 times compression. This is because the student model is trained with an linear interpolation of hard probabilities and soft probabilities. If some labels of the student model have errors, the teacher model may eliminate some of these errors. Meanwhile, if some probabilities of the teacher model are incorrect, the student model may correct errors. However, the method KL [16] is proposed to train the student model only with soft probabilities. Therefore, our method is more effective.

7 Conclusion

This paper proposes a method to compress large acoustic models with knowledge transfer for mobile devices. The small student model is trained with a linear interpolation of hard probabilities and soft probabilities. Thus, the student model can learn the generalized knowledge from the teacher model. If some labels of the student model have errors, the teacher model may eliminate some of these errors and vice-versa. The experiments on RASC863 and AMI corpus show that our proposed method can obviously compress acoustic models without performance loss. In future work, we plan to reduce the parameters of the output layer.

References


