



Accurate and Structured Pruning for Efficient Automatic Speech Recognition

Huiqiang Jiang¹, Li Lyna Zhang¹, Yuang Li², Yu Wu¹, Shijie Cao¹,
Ting Cao¹, Yuqing Yang¹, Jinyu Li³, Mao Yang¹, Lili Qiu¹

¹Microsoft Research, ²University of Cambridge, ³Microsoft Azure Speech
{hjiang, lzhani, wu.yu, shjiecao, ting.cao, yuqing.yang, jinyuli, maoyang,
liliqui}@microsoft.com, yl807@eng.cam.ac.uk

Abstract

Automatic Speech Recognition (ASR) has seen remarkable advancements with deep neural networks, such as Transformer and Conformer. However, these models typically have large model sizes and high inference costs, posing a challenge to deploy on resource-limited devices. In this paper, we propose a novel compression strategy that leverages structured pruning and knowledge distillation to reduce the model size and inference cost of the Conformer model while preserving high recognition performance. Our approach utilizes a set of binary masks to indicate whether to retain or prune each Conformer module, and employs L_0 regularization to learn the optimal mask values. To further enhance pruning performance, we use a layer-wise distillation strategy to transfer knowledge from unpruned to pruned models. Our method outperforms all pruning baselines on the widely used LibriSpeech benchmark, achieving a 50% reduction in model size and a 28% reduction in inference cost with minimal performance loss.

Index Terms: ASR, Model Compression, Structured Pruning

1. Introduction

Large models such as Conformer [1], wav2vec 2.0 [2], and Whisper [3] have achieved remarkable success in Automatic Speech Recognition (ASR) systems, yet they come at a high cost in terms of storage, memory, and inference latency. For instance, the Whisper large model comprises 1.5 billion parameters, making it extremely difficult to deploy on resource-constrained scenarios such as client edge devices. Therefore, it is crucial to compress ASR models for practical deployment.

Knowledge distillation [4, 5] have been demonstrated to be effective in producing a faster model for ASR [6, 7, 8, 9]. However, these methods have several limitations. First, it necessitates a meticulous design to customize the high-quality smaller model [10, 11], which requires expertise and can be expensive, given the various deployment requirements of real-world ASR applications. Second, distillation involves a full training from scratch process [12, 13], which is time-consuming.

Weight pruning can efficiently produce a small model that meets a given resource constraint. In particular, magnitude pruning [14, 15], which preserves model parameters with high absolute values, is the most widely used method for pruning an ASR model [16, 17, 18]. However, it can often lead to unsatisfactory performance under a high compression ratio. Moreover, while these weight pruning methods enable the removal of weights at arbitrary locations, resulting in sparsity, these sparse models are usually unstructured, which hinders actual efficiency benefits. This is because running them in standard hardware usually requires reconstructing the original dense shape, leading to little improvement in latency [16, 19].

In this work, we propose a novel structured pruning approach to directly speed up ASR inference of a Conformer-Transducer [1] model, while preserving high accuracy. Our approach leverages learnable binary masks to determine whether to retain or prune distinct components of the Conformer encoder within a hybrid pruning granularity, including: (i) an entire attention head in a multi-head self-attention layer, (ii) a specific dimension in the intermediate layer of the feed-forward network (FFN), (iii) the entire convolution module in each Conformer block, and (iv) the hidden dimension. To determine the values of the masks, we employ an augmented L_0 regularization approach that makes the discrete masks differentiable. In contrast to magnitude pruning, our approach jointly learns the masks and updates model parameters to achieve optimal pruning decisions. Furthermore, we introduce a layerwise knowledge distillation technique to further enhance the pruning decisions and performance by transferring layerwise hidden states from the unpruned model to the pruned model.

We evaluate our method on the widely-used LibriSpeech benchmark [20] under various compression ratios. Experimental results show that our method can achieve up to 50% parameter pruning with minimal degradation in performance for Conformer-Transducer. Furthermore, the pruned model reduces the real-time factor by 28% on a CPU device.

2. Method

In this section, we present our approach for expediting ASR inference on hardware resource-limited scenarios. We use Conformer [1] as the representative model due to its wide usage in ASR. Our approach consists of two steps. First, we introduce a hybrid structured pruning strategy tailored to the CNN-Transformer architecture of the Conformer model, which assigns binary masks to various modules such as attention heads and hidden dimensions to indicate their removal or retention. Second, we leverage the power of knowledge distillation and incorporate an L_0 regularization pruning algorithm to jointly optimize the binary pruning masks and the model parameters in order to strike the best balance between speed and accuracy.

2.1. Hybrid pruning granularity

The model pruning problem can be formulated as:

$$\min_{\theta, \mathbf{z}} \mathbb{E}_{\mathbf{z}} \left[\frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\text{task}}(\mathbf{x}_i, \mathbf{y}_i; \theta \odot \mathbf{z}) + \lambda \|\mathbf{z}\|_0 \right] \quad (1)$$

where θ denotes the model weights, $\mathbf{z} \in \{0, 1\}$ is the binary mask introduced by pruning, where 0 indicates removal and 1 indi-

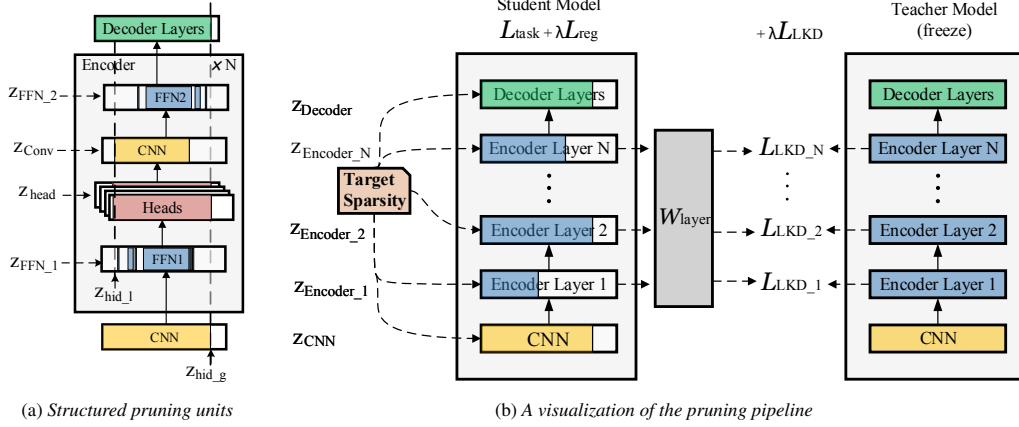


Figure 1: (a) the hybrid pruning granularity designed for Conformer model; (b) The overview of our pruning algorithm. By combining knowledge distillation and pruning, our approach learns an optimal set of pruning masks. W_{layer} is the spatial transformation matrix in layer-wise knowledge distillation (Eq. 8).

icates retention of a corresponding parameter. N , \mathbf{x}_i , \mathbf{y}_i and \mathcal{L}_{task} are the sample size, input, labels, and task loss of ASR task. λ is the hyperparameter that controls the tradeoff between accuracy and sparsity. $\|\cdot\|_0$ denotes the 0-norm loss coefficient.

Previous research [16, 17] has mostly employed magnitude-based unstructured pruning, which allows the pruning of unimportant parameters at arbitrary locations based on weights value. However, this method often yields poor performance with higher levels of sparsity and does not directly offer acceleration benefits. Recently, unstructured [21, 22] and structured [23, 24, 25] pruning methods have been proposed in other domains. However, these methods are not specifically designed for ASR tasks and may result in suboptimal performance.

To address this, our objective is to structurally remove unimportant parameters for ASR models, thereby enabling faster inference on standard hardware while preserving accuracy. While prior research has primarily focused on pruning transformer or CNN models, our approach is tailored to the Conformer architecture, which features a unique combination of convolution and transformer modules. To achieve optimal results, we must develop a specific pruning policy that accounts for all four modules that comprise each Conformer encoder layer: a feed-forward module (FFN₁), a multi-head self-attention module (MHA), a convolution module, and a second feed-forward module (FFN₂). At the same time, we also prune the embedding layers and the decoding layers. To attain this goal, we propose four types of binary masks $\mathbf{z} \in \{0, 1\}$ to control the sparsity of different modules, as illustrated in Fig. 1a.

- **HEAD MASK \mathbf{z}_{HEAD} .** We use $\mathbf{z}_{head_j}^i$ to determine whether j -th head of the i -th encoder layer should be kept or removed.
- **FFN INTERMEDIATE MASK \mathbf{z}_{FFN} .** We use masks $\mathbf{z}_{FFN_j}^i$ to prune the i -th FFN j -th channel in intermediate dimensions.
- **CONV MODULE MASK \mathbf{z}_{CONV} .** Although the convolution module has comparatively few parameters, it contributes significantly to the total number of floating-point operations (FLOPs) in the model. To maximize inference efficiency, we introduce a gate mask, denoted as \mathbf{z}_{conv}^i , which allows for the pruning of the i -th entire convolution module in each block.
- **LAYER-WISE HIDDEN MASK \mathbf{z}_{HIDDEN} .** We prune the hidden dimensions of the encoder layers to enable more flexibility by using global and local masks, denoted as \mathbf{z}_{hidden} . The global mask, \mathbf{z}_{hid-g} , determines the size of hidden represen-

tations across all encoder layers, while the local mask, \mathbf{z}_{hid-1} , controls the hidden size within an encoder layer.

2.2. Learning the optimal mask under a desired sparsity

Based on the hybrid pruning granularity proposed in Section 2.1, this paper designs an L_0 -based pruning method (as shown in Fig. 1b) that gradually reaches a target sparsity during the pruning training process and learns the optimal sparsity allocation according to the ASR task loss. Moreover, we combine a layer-wise knowledge distillation with pruning to further enhance the performance.

Sparsity-aware Constraint. With the use of pruning masks, we can measure the number of retained model parameters and, thus, calculate the inference cost. Formally, the expected model parameters after pruning can be calculated as follows:

$$S(\mathbf{z}) = \sum_{i=1}^L [S(\mathbf{z})_{FFN_1}^i + S(\mathbf{z})_{Attn}^i + S(\mathbf{z})_{CNN}^i + S(\mathbf{z})_{FFN_2}^i] \quad (2)$$

where L is the number of encoder layers and $S(\mathbf{z})^i$ denote the parameters of different modules in i -th layer after pruning. We calculate the retained parameters of each module as follows:

$$\begin{aligned} S(\mathbf{z})_{FFN_1}^i &= \|\mathbf{z}_{FFN_1}^i\|_0 \times (2 \times \|\mathbf{z}_{i,hid-g}\|_0 + \|\mathbf{z}_{i,hid-1}\|_0) \\ S(\mathbf{z})_{Attn}^i &= \|\mathbf{z}_{hid-1}^i\|_0 \times \|\mathbf{z}_{head}^i\|_0 \times \mathbb{S}_{head} \\ S(\mathbf{z})_{CNN}^i &= \|\mathbf{z}_{hid-1}^i\|_0 \times \|\mathbf{z}_{conv}^i\|_0 \times \mathbb{S}_{conv} \\ S(\mathbf{z})_{FFN_2}^i &= \|\mathbf{z}_{FFN_2}^i\|_0 \times (2 \times \|\mathbf{z}_{i,hid-1}\|_0 + \|\mathbf{z}_{i,hid-g}\|_0) \end{aligned} \quad (3)$$

Similarly, the FLOPs value $S(\mathbf{z})_{FLOPs}$ of the pruned model can be obtained, thus providing an estimation of the computational cost that would be required on edge devices.

Sparse Allocation Optimization based on L_0 Regularization. Now, we introduce a method for determining the values of masks for pruning. Unlike magnitude pruning, which sets masks based on the values of weights, our proposed approach seeks to jointly learn masks and model parameters so that the resulting model can achieve minimal task empirical risk while meeting the desired sparsity. To achieve this, a L_0 regularization term \mathcal{L}_{reg} [26] is added to the optimization problem, turning the pruning problem into an end-to-end learning problem.

$$\mathcal{L} = \mathcal{L}_{task}(\theta, \mathbf{z}) + \lambda \mathcal{L}_{reg}(\mathbf{z}) \quad (4)$$

However, these binary masks \mathbf{z} are discrete values and non-differentiable [27]. To address this challenge, we use the reparameterization method with the hard concrete distribution proposed by [26]. Specifically, masks \mathbf{z} can be sampled from a random variable $\mathbf{u} \sim U(0, 1)$ distribution:

$$\mathbf{t} = \text{sigmoid}(\log \frac{\mathbf{u}}{1 - \mathbf{u}} + \alpha); \mathbf{z} = \min(1, \max(0, \mathbf{t} \times (r - l) + l)) \quad (5)$$

Where $l < 0$ and $r > 0$ are two constants, with the common practice of setting l to -0.1 and r to 1.1 , parameters α are learnable. Then, the L_0 regularization term can be regulated as a function of the cumulative density function $Q(\mathbf{t} \leq 0|\alpha)$, which will be differentiable. The expectation of mask \mathbf{z} in Eq. 1 can be expressed as,

$$\mathbb{E}_{\mathbf{z}}\|\mathbf{z}\|_0 = 1 - Q(\mathbf{t} \leq 0|\alpha) = 1 - \text{sigmoid}(\log \frac{\mathbf{t} - l}{r - \mathbf{t}} - \alpha) \quad (6)$$

Next, we augment L_0 regularization to better control the achieved model sparsity. The λ in Eq. 4 is used to balance the trade-off between model accuracy and sparsity. However, it requires careful hyper-parameter tuning to make sure it converges to a desired sparsity. To effectively control the final model sparsity, we follow [28] and replace the original L_0 regularization with a Lagrangian multiplier. Let s denote the target model size after pruning, and $S(\mathbf{z})$ denote the expected model parameters determined by the masks \mathbf{z} in Eq. 2. We impose an equality constraint $S(\mathbf{z}) = s$ by introducing a violation penalty:

$$\mathcal{L}_{reg}(\mathbf{z}) = \lambda_1(S(\mathbf{z}) - s) + \lambda_2(S(\mathbf{z}) - s)^2 \quad (7)$$

where λ_1 and λ_2 are automatically adjusted using the AdamW optimizer [29].

Combining Pruning and Knowledge Distillation (KD). The combination of knowledge distillation and pruning has been shown to perform better than pruning alone [21, 19]. Previous works typically adopt a cross-entropy distillation loss between the teacher and student models in the final prediction layer. Our method, however, employs a layer-by-layer distillation to best utilize the layerwise semantic information in the teacher model. Through this guidance, the pruned model is encouraged to preserve more detailed layerwise semantic knowledge, leading to improved accuracy.

Specifically, we use the original unpruned model as the teacher and the model under pruning as the student. For a Conformer model with L encoder layers, our layerwise knowledge distillation aims to minimize the average distance between the hidden states of each encoder layer of the teacher and student models, measured by the Mean Squared Error (MSE) loss.

To this end, the full training objective is a combination of the aforementioned objectives:

$$\mathcal{L} = \mathcal{L}_{\text{task}}(\theta, \mathbf{z}) + \lambda \mathcal{L}_{reg}(\mathbf{z}) + \frac{1}{L} \sum_{i=1}^L \mathcal{L}_{\text{MSE}}(\mathbf{h}_{\text{stu}}^i, \mathbf{W}_{\text{layer}} \cdot \mathbf{h}_{\text{tea}}^i) \quad (8)$$

where $\mathbf{h}_{\text{tea}}^i$ and $\mathbf{h}_{\text{stu}}^i$ represent the i -th layer hidden outputs of the teacher and student models, respectively. $\mathbf{W}_{\text{layer}}$ denotes the transformation matrix between $\mathbf{h}_{\text{tea}}^i$ and $\mathbf{h}_{\text{stu}}^i$, which can improve the distillation performance.

3. Experiments

3.1. Experimental Setup

Model and dataset. We apply our pruning method on the Conformer-Transducer model and evaluate its performance on

Table 1: The WER results in the LibriSpeech test set varied across different sparsity targets. Our results (both sparsity-aware and FLOPs-aware) are represented by “Ours (Sparsity)” and “Ours (FLOPs)”, respectively.

Sparsity	Methods	Test-clean	Test-other	RTF	Model Size (MB)
0%	Conformer	2.86	6.10	0.193	75.5
20%	OMP [16]	3.12	6.41	0.174 (-9.7%)	61.9 (-18.1%)
	PARP [16]	3.12	6.24	0.175 (-9.1%)	61.9 (-18.1%)
	SVD [30]	4.23	8.59	0.180 (-6.4%)	60.3 (-20.2%)
	Ours (Sparsity)	2.96	6.09	0.171 (-11.3%)	60.2 (-20.3%)
	Ours (FLOPs)	2.88	6.14	0.174 (-9.9%)	61.6 (-18.4%)
30%	OMP [16]	3.28	6.99	0.164 (-14.9%)	53.8 (-28.7%)
	PARP [16]	3.31	6.83	0.165 (-14.2%)	53.8 (-28.7%)
	SVD [30]	4.70	9.69	0.171 (-11.2%)	52.7 (-30.2%)
	Ours (Sparsity)	3.05	6.29	0.164 (-14.8%)	52.1 (-31.0%)
	Ours (FLOPs)	3.08	6.41	0.161 (-16.2%)	47.3 (-37.4%)
40%	OMP [16]	3.56	7.55	0.152 (-21.0%)	45.8 (-39.3%)
	PARP [16]	3.54	7.39	0.153 (-20.8%)	45.8 (-39.3%)
	SVD [30]	4.75	9.76	0.163 (-15.6%)	45.2 (-40.2%)
	Ours (Sparsity)	3.12	6.61	0.152 (-21.1%)	44.1 (-41.6%)
	Ours (FLOPs)	3.15	6.88	0.152 (-21.1%)	40.0 (-47.1%)
50%	OMP [16]	3.57	8.02	0.145 (-24.8%)	37.8 (-50.0%)
	PARP [16]	3.56	7.83	0.147 (-23.7%)	37.8 (-50.0%)
	SVD [30]	4.78	9.84	0.151 (-21.6%)	37.7 (-50.1%)
	Ours (Sparsity)	3.27	6.89	0.144 (-25.0%)	37.1 (-50.9%)
	Ours (FLOPs)	3.29	7.17	0.139 (-28.1%)	36.7 (-51.4%)

the standard LibriSpeech dataset [20], which contains 960 hours of training speech data. Specifically, the model consists of 18 encoder layers and 2 decoder layers. Each encoder layer has 512 hidden dimensions, 8 heads, 1024 FFN intermediate dimensions, and convolution with kernel size of 3. In our work, we only mask and prune modules in encoder layers.

Pruning. Before pruning, we follow the training receipts in Conformer[1] to pre-train the model until convergence. Then, we set various sparsity ratios of 20%, 30%, 40%, 50% and apply our approach on the pre-trained model under each sparsity ratio. We consider both the FLOPs sparsity ratio and model parameter size ratio as discussed in Sec. 2.2. To learn the pruning masks, we utilize three AdamW [29] optimizers, one for the ASR task transducer loss, another for the L_0 loss, and the third for the Lagrangian constraint loss. The initial learning rates are set to $1e-2$, $1e-2$, and $3e-4$, respectively. We set λ to 1. For each sparsity ratio, we gradually increase the sparsity from 0 to the target value within the first 10k steps, and then keep the target value for the remaining 100k steps. All the experiments are trained on 8 NVIDIA V100 GPUs using ESPnet¹.

Baselines. We compare our approach with other widely-used compression methods: SVD [30] and two magnitude pruning methods (One-shot Magnitude Pruning (OMP), PARD [16]). In particular, OMP and PARD are originally implemented for unstructured pruning, which do not provide any benefits to inference latency. For a fair comparison, we implement OMP and PARD using our structured pruning granularity setting.

To evaluate the performance of the pruned models, we report the WER results on the LibriSpeech test, as well as the encoder model size and the inference real-time factor (RTF) of the encoder part on a single-core Intel(R) Xeon(R) CPU (2.60GHz).

3.2. Main Results

Table 1 presents a summary of the WER results as well as RTF achieved by various methods. To begin with, we compare the performance of our pruned models with the original Conformer model (0%). Notably, we are able to remove 20% of the unim-

¹<https://github.com/espnet/espnet>

Table 2: The WER results of different ablation settings on LibriSpeech test set. “w/o layer-wise hidden masks” means we remove layer-wise hidden dimension pruning and “w head pruning” denotes only conducting head pruning.

Sparsity	Methods	Test-clean	Test-other
20%	Ours (Sparsity)	2.96	6.09
	w/o layer-wise hidden	2.98	6.24
	w head level pruning	3.14	6.69
	w/o layer-wise KD	2.95	6.19
	Ours (Sparsity)	3.05	6.29
30%	w/o layer-wise hidden	3.12	6.45
	w head level pruning	3.37	7.03
	w/o layer-wise KD	3.07	6.30
	Ours (Sparsity)	3.12	6.61
	w/o layer-wise hidden	3.26	6.89
40%	w head level pruning	3.58	7.39
	w/o layer-wise KD	3.15	6.73
	Ours (Sparsity)	3.27	6.89
	w/o layer-wise hidden	3.41	7.19
	w head level pruning	3.72	7.82
50%	w/o layer-wise KD	3.29	6.93

portant model parameters from the Conformer model without any significant loss in WER performance. This results in an 11.3% inference acceleration. As the sparsity increases, we observe a slight drop in WER performance but with a notable improvement in inference acceleration. Interestingly, even at a high sparsity ratio of 50%, our pruned model still achieves better performance than the open-source Conformer², with noteworthy WER of 3.27 and 6.89 on the test-clean and test-other datasets, respectively. This translates into a 25% reduction in RTF, demonstrating the effectiveness of our pruning approach.

Table 1 clearly demonstrates that our method outperforms all the baseline state-of-the-art structured pruning methods, achieving significantly better results at all sparsity ratios. SVD baseline exhibits poor performance on the LibriSpeech dataset, which is known to be particularly sensitive to model parameters. When compared to the magnitude-based method, our approach incurs smaller WER losses, especially on the test-other dataset, with relative WER loss percentages of 8.23%/5.25%, 7.91%/11.60%, 15.23%/15.36%, and 10.47%/18.55% at 20%, 30%, 40%, and 50% sparsity targets, respectively. Compared to PARD, our approach achieves relative WER loss improvements of 8.30%/2.56%, 9.21%/8.98%, 14.74%/12.82%, and 10.01%/15.42% on the test-clean and test-other datasets, respectively. Notably, as the sparsity targets increase, our method achieves a greater advantage in terms of WER loss.

3.3. Ablation Study

We now conduct ablation studies to evaluate (1) the effectiveness of our hybrid pruning granularity and (2) the effectiveness of combining knowledge distillation with pruning. For experiment (1), we compare our current setting with head pruning (w head level pruning), which is widely-used in other NLP tasks [31, 32], and we compare with disabling the pruning for layer-wise hidden dimension (w/o layer-wise hidden). For experiment (2), we remove layer-wise KD for evaluation.

Table 2 demonstrates the effectiveness of our proposed hybrid pruning approach, which enables us to selectively prune fine-grained units such as heads, Conv, FFN, and layer-wise hidden dimensions. In contrast, conducting head pruning alone leads to a significant loss of 32.9% in WER because crucial heads can be removed to achieve high sparsity. Disabling layer-wise hidden pruning also slightly degrades the performance.

²<https://github.com/espnet/espnet/blob/master/egs/librispeech/asr1>

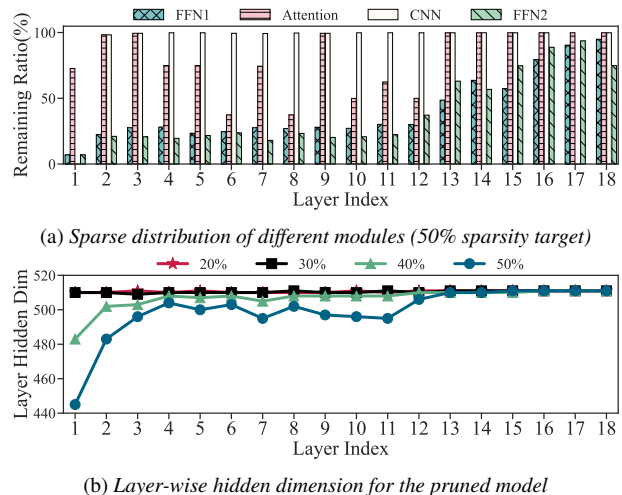


Figure 2: The distribution of the remaining ratio across different modules and layers when aiming for different sparsity targets.

Table 2 also suggests that the proposed combination of knowledge distillation and pruning effectively improves the ASR performance. It is worth noting that layer-wise KD consistently demonstrates an absolute WER improvement of 0.04-0.13 under different sparse targets. This can be attributed to the fact that layer-wise KD facilitates the transfer of hierarchical information from the teacher model to the student model on a layer-by-layer basis, which encourages the pruned model to retain more semantic knowledge.

3.4. Analysis of Pruning Results

Finally, we study the pruned Conformer structures produced by our approach. Fig. 2a shows the retained parameters ratios for the 50% sparsity model. Interestingly, the Conformer model shows more redundancy in the bottom layers. Specifically, our approach removes up to 76.7% parameters at the bottom layers (Layer1-10), while only 6.08%-11.9% are removed at the top layers (Layer 16-18). The results suggest that the Conformer-Transducer model exhibits a bottom-heavy redundancy and top-sensitive pattern in the ASR task. Moreover, the model displays varied redundancy in different modules. Compared to attention and CNN modules, we prune more parameters in FFN layers.

Fig. 2b shows the remaining hidden dimensions of various layers at different sparsity ratios. As the target sparsity ratio increases, more hidden dimensions are pruned. Notably, at the target sparsity of 50%, the lower layers can prune up to 15% of their dimensions, while deeper layers exhibit a very minimal pruning. These findings suggest that deeper layer model parameters are crucial for ASR performance, and even slight changes in their dimension sizes can have a significant impact.

4. Conclusion

In this paper, we propose a structured pruning algorithm that boosts the efficiency of automatic speech recognition by pruning unimportant modules with a hybrid granularity. We target all levels of pruning in the Conformer model, including Attention heads, FFN layers, Conv module, and layer-wise hidden dimension. Our approach combines L_0 regularization and knowledge distillation to achieve optimal pruning decisions. Experiments on LibriSpeech show that our method cuts model size by 50% and CPU inference cost by 28%, with a minimal WER performance loss, far exceeding the baseline model.

5. References

- [1] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented transformer for speech recognition,” in *Proc. Interspeech 2020*, H. Meng, B. Xu, and T. F. Zheng, Eds. ISCA, 2020, pp. 5036–5040.
- [2] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Proc. NeurIPS 2020*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.
- [3] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” *ArXiv preprint*, vol. abs/2212.04356, 2022.
- [4] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, “Learning small-size DNN with output-distribution-based criteria,” in *Proc. Interspeech*, 2014, pp. 1910–1914.
- [5] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.
- [6] K.-P. Huang, Y.-K. Fu, T.-Y. Hsu, F. R. Gutierrez, F.-L. Wang, L.-H. Tseng, Y. Zhang, and H.-y. Lee, “Improving generalizability of distilled self-supervised speech processing models under distorted settings,” in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 1112–1119.
- [7] Q. Xu, T. Song, L. Wang, H. Shi, Y. Lin, Y. Lv, M. Ge, Q. Yu, and J. Dang, “Self-distillation based on high-level information supervision for compressing end-to-end asr model,” *Proc. Interspeech 2022*, pp. 1716–1720, 2022.
- [8] J. Rathod, N. Dawalatabad, S. SINGH, and D. Gowda, “Multi-stage Progressive Compression of Conformer Transducer for On-device Speech Recognition,” in *Proc. Interspeech 2022*, 2022, pp. 1691–1695.
- [9] Z. Wei, L. Hao, and X. Zhang, “Model compression by iterative pruning with knowledge distillation and its application to speech enhancement,” *Proc. Interspeech 2022*, pp. 941–945, 2022.
- [10] H.-J. Chang, S.-w. Yang, and H.-y. Lee, “Distilhubert: Speech representation learning by layer-wise distillation of hidden-unit bert,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7087–7091.
- [11] Y. Meng, H.-J. Chen, J. Shi, S. Watanabe, P. Garcia, H.-y. Lee, and H. Tang, “On compressing sequences for self-supervised speech models,” in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 1128–1135.
- [12] Y. Wang, C. Tang, Z. Ma, Z. Zheng, X. Chen, and W.-Q. Zhang, “Exploring effective distillation of self-supervised speech models for automatic speech recognition,” *ArXiv preprint*, vol. abs/2210.15631, 2022.
- [13] Y. Lee, K. Jang, J. Goo, Y. Jung, and H. R. Kim, “FitHuBERT: Going Thinner and Deeper for Knowledge Distillation of Speech Self-Supervised Models,” in *Proc. Interspeech 2022*, 2022, pp. 3588–3592.
- [14] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” *International Conference on Learning Representations (ICLR)*, 2016.
- [15] S. Han, J. Pool, J. Tran, and W. J. Dally, “Learning both weights and connections for efficient neural network,” in *Proc. NeurIPS 2015*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 1135–1143.
- [16] C.-I. J. Lai, Y. Zhang, A. H. Liu, S. Chang, Y.-L. Liao, Y.-S. Chuang, K. Qian, S. Khurana, D. Cox, and J. Glass, “Parp: Prune, adjust and re-prune for self-supervised speech recognition,” *Proc. Interspeech 2020*, vol. 34, pp. 21 256–21 272, 2021.
- [17] J. Kim, S. Chang, and N. Kwak, “PQK: Model Compression via Pruning, Quantization, and Knowledge Distillation,” in *Proc. Interspeech 2021*, 2021, pp. 4568–4572.
- [18] V. S. Lodagala, S. Ghosh, and S. Umesh, “Pada: Pruning assisted domain adaptation for self-supervised speech representations,” in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 136–143.
- [19] F. Lagunas, E. Charlaix, V. Sanh, and A. Rush, “Block pruning for faster transformers,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021, pp. 10 619–10 629.
- [20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*. IEEE, 2015, pp. 5206–5210.
- [21] V. Sanh, T. Wolf, and A. M. Rush, “Movement pruning: Adaptive sparsity by fine-tuning,” in *Proc. NeurIPS 2020*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.
- [22] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, “Importance estimation for neural network pruning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 264–11 272.
- [23] F. Lagunas, E. Charlaix, V. Sanh, and A. Rush, “Block pruning for faster transformers,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 10 619–10 629.
- [24] M. Xia, Z. Zhong, and D. Chen, “Structured pruning learns compact and accurate models,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1513–1528.
- [25] L. L. Zhang, Y. Homma, Y. Wang, M. Wu, M. Yang, R. Zhang, T. Cao, and W. Shen, “Swiftpruner: Reinforced evolutionary pruning for efficient ad relevance,” in *Proceedings of the 31st ACM International Conference on Information Knowledge Management*, ser. CIKM ’22, 2022, p. 3654–3663.
- [26] C. Louizos, M. Welling, and D. P. Kingma, “Learning sparse neural networks through l0 regularization,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.
- [27] Y. Bengio, N. Léonard, and A. Courville, “Estimating or propagating gradients through stochastic neurons for conditional computation,” *arXiv preprint arXiv:1308.3432*, 2013.
- [28] Z. Wang, J. Wohlwend, and T. Lei, “Structured pruning of large language models,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020, pp. 6151–6162.
- [29] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.
- [30] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, “Semi-orthogonal low-rank matrix factorization for deep neural networks,” in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, B. Yegnanarayana, Ed. ISCA, 2018, pp. 3743–3747.
- [31] P. Michel, O. Levy, and G. Neubig, “Are sixteen heads really better than one?” in *Proc. NeurIPS 2019*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 14 014–14 024.
- [32] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, “Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 5797–5808.