Interspeech 2024
1-5 September 2024, Kos, Greece

Finding Task-specific Subnetworks in Multi-task Spoken Language Understanding Model

Hayato Futami[1], *Siddhant Arora*[2], *Yosuke Kashiwagi*[1], *Emiru Tsunoo*[1], *Shinji Watanabe*[2]

[1]Sony Group Corporation, Japan
[2]Carnegie Mellon University, USA
Hayato.Futami@sony.com

Abstract

Recently, multi-task spoken language understanding (SLU) models have emerged, designed to address various speech processing tasks. However, these models often rely on a large number of parameters. Also, they often encounter difficulties in adapting to new data for a specific task without experiencing catastrophic forgetting of previously trained tasks. In this study, we propose finding task-specific subnetworks within a multi-task SLU model via neural network pruning. In addition to model compression, we expect that the forgetting of previously trained tasks can be mitigated by updating only a task-specific subnetwork. We conduct experiments on top of the state-of-the-art multi-task SLU model "UniverSLU", trained for several tasks such as emotion recognition (ER), intent classification (IC), and automatic speech recognition (ASR). We show that pruned models were successful in adapting to additional ASR or IC data with minimal performance degradation on previously trained tasks.

Index Terms: spoken language understanding, speech recognition, network pruning, continual learning

1. Introduction

Many recent studies on language and speech processing have been advancing toward unified models that can solve a wide range of tasks. Large Language Models (LLMs) [1] are epitomes of those. LLMs have the ability to perform various natural language processing (NLP) tasks, instructed by prompts that describe the task. In the field of speech processing, SpeechPrompt [2, 3] has proposed prompt tuning on Generative Spoken Language Model (GSLM) [4] to perform a wide variety of spoken language understanding (SLU) tasks. SLU, which we focus on in this study, covers understanding semantics, paralinguistics, and content from speech [1]. We include a few representative SLU tasks in this study, such as intent classification (IC), emotion recognition (ER), and automatic speech recognition (ASR). Recently, UniverSLU has been proposed as a universal spoken language understanding (SLU) model, which fine-tunes Whisper [6] by extending Whisper-style task specifier to various SLU tasks [5]. UniverSLU has shown superior performances compared to the state-of-the-art task-specific models and prior unified models. More recently, several studies have been working on extending LLMs to process audio input, performing some SLU tasks [7, 8, 9].

Although such unified multi-task models are promising in performance and usability, they often suffer from large model sizes. Another problem is that when new data for a specific task become available, additional training on the task can lead to catastrophic forgetting [10] of other tasks trained before. This is addressed by continual learning, which has been getting attention in ASR [11, 12, 13, 14, 15] and SLU [16, 17]. Continual learning is especially in demand for large-scale speech foundation models, as they require high retraining cost [18].

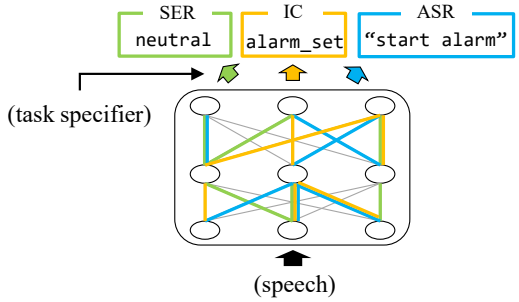


Figure 1: *Illustration of task-specific subnetworks in multi-task SLU model. To solve SER task, only subnetwork represented as green pathways is activated.*

To solve the above two issues, we propose a network pruning method to find task-specific subnetworks in a multi-task SLU model. Pruning has been applied to ASR and SLU to reduce memory footprint and speedup in inference [19, 20, 21, 22]. A recent study on multi-lingual ASR [23] has succeeded in finding language-specific subnetworks via Lottery Ticket Hypothesis [24] -based pruning. In this study, we extend it to multi-task SLU, where pruning is applied to identify task-specific subnetworks defined as pathways on the model. During training and inference of a task, only a subnetwork for the task is activated, as shown in Figure 1. While pruning reduces parameter counts, we also expect better continual learning via network pruning, not investigated in [23]. During training, only parameters in the task-specific subnetwork are updated while others remain unchanged, which works to mitigate catastrophic forgetting of other tasks.

In this study, we focus on finding task-specific subnetworks in UniverSLU [5]. We have trained the model for ER on IEMOCAP corpus [25], IC on SLURP [26], and ASR on LibriSpeech [27]. We found that pruned models achieved better performances on ER and IC with much smaller parameter counts. We also explore continual learning on additional ASR or IC data. We observed that, for pruned models, the performances of tasks not currently trained were less deteriorated. We further observed that the method was effective for UniverSLU trained on more tasks including named entity recognition (NER), and speech command recognition (SCR), and IC on other datasets, which also offers insight into how similarity between tasks lead to overlap between task-specific subnetworks.

[1]We use the term SLU in a broader sense, following [2, 3, 5]

10.21437/Interspeech.2024-712

2. Related work

2.1. Network pruning

Network pruning is a technique for removing unnecessary parameters from neural networks, which can reduce model size and computation costs. Structured pruning remove parameters in groups, while unstructured pruning remove them individually [28]. This study focuses on unstructured pruning, where it is less effective to achieve computational efficiency in modern libraries and hardware. However, both often share the same insights, and these insights can be applied to each other, as indicated in previous studies [19, 24]. The Lottery Ticket Hypothesis (LTH) study [24] has demonstrated the existence of sparse subnetworks (called "winning tickets") that match or even surpass the performance of the original dense networks. There are several recent studies on pruning in speech processing [19, 20, 21, 22, 23]. In [20], LTH for ASR has been investigated. Pruning of self-supervised learning (SSL) based models have been investigated in [19] and [21]. Some other studies focus on obtaining multiple subnetworks of different properties within a single model [22, 23]. Omni-sparsity DNN [22] trains subnetworks of different sparsity. In [23], language-specific subnetworks in a multi-lingual ASR model are considered. Unlike these studies, our study focuses on a multi-task SLU model and looks for subnetworks for different SLU tasks. Moreover, our study represents the first investigation into the benefits of continual learning via pruning in SLU.

2.2. Continual learning

Continual learning, or lifelong learning, aims to learn new data while preventing catastrophic forgetting of previously learned knowledge. Continual learning methods can be categorized into three: regularization-based, replay-based, and architecture-based methods. Regularization-based methods introduce an additional regularization term [29, 30], which has also been investigated in ASR [11] and SLU [16, 17]. Replay-based methods replay examples from previous data [31], including studies for ASR [12]. Architecture-based methods spare isolated parameters for new task. This can be done by updating only a part of the entire parameters [13, 14, 15]. Especially in [32] and [33], network pruning is applied sequentially for image recognition tasks, where parameters not used in preceding tasks are allocated for the current task. In contrast, parameter sharing between tasks can exist in our study. This is more parameter efficient and flexible, where each subnetwork can be designed with any sparsity.

3. Finding task-specific subnetworks

In this study, we propose finding task-specific subnetworks in a multi-task SLU model. We aim to identify individual pathways for each SLU task on a single dense network of shared parameters, which we call *multi-task pruning*. On the other hand, obtaining individual sparse models for SLU tasks is called *single-task pruning*. Multi-task pruning allows us to switch tasks by switching only pathways, which is parameter efficient and preferable for deployment. To realize this, we adapt a pruning method originally proposed for multi-lingual ASR [23] to multi-task SLU.

Let $f(X, s_t; \theta)$ denote a multi-task SLU network with input speech X, prompt of task specifier s_t, and parameters θ. Task specifier is defined for each task $t \in \mathcal{T}$ and instruct the network which task to solve, where $\mathcal{T}$ denotes a set of SLU tasks.

Algorithm 1 Identify pruning mask

1: Initialize $f(X, s_t; m_t \odot \theta)$ with $\theta \leftarrow \theta_0$, $m_t \leftarrow \{1\}^{|\theta|}$.
2: **repeat**
3: **for** $t \in \mathcal{T}$ **do**
4: **repeat**
5: Update $\theta \leftarrow \mathrm{TrainNetwork}(f(X, s_t; m_t \odot \theta), \mathcal{D}_t)$
6: **until** The loop repeated N_1 times
7: Update $m_t \leftarrow \mathrm{Prune}(\theta, m_t, p)$
8: Reset $\theta \leftarrow \theta_0$.
9: **end for**
10: **until** The loop repeated Q times

Algorithm 2 Update parameters

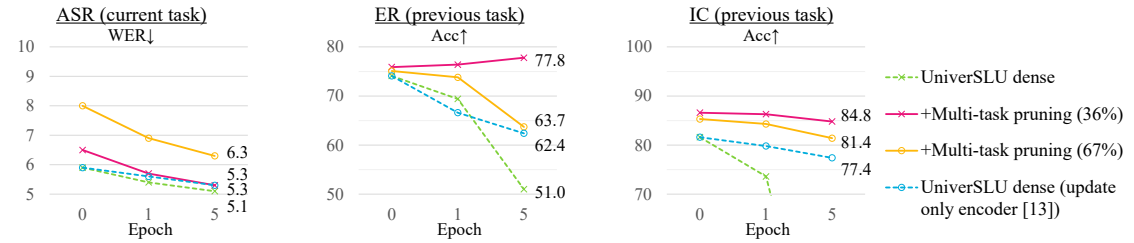
1: Initialize $f(X, s_t; m_t \odot \theta)$ with $\theta \leftarrow \theta_0$, m_t from Algo.1.
2: **repeat**
3: **for** $t \in \mathcal{T}$ **do**
4: **repeat**
5: Update $\theta \leftarrow \mathrm{TrainNetwork}(f(X, s_t; m_t \odot \theta), \mathcal{D}_t)$
6: **until** The loop repeated N_2 times
7: **end for**
8: **until** The loop repeated R times

For pruning, we reformulate $f(X, s_t; \theta)$ as $f(X, s_t; m \odot \theta)$, where $m \in \{0, 1\}^{|\theta|}$ denotes a pruning mask. We assume task-specific pruning, where m is task-specific denoted as m_t for task t. Our multi-task pruning consists of two steps: (1) identifying pruning mask m_t and (2) updating parameters θ using m_t.

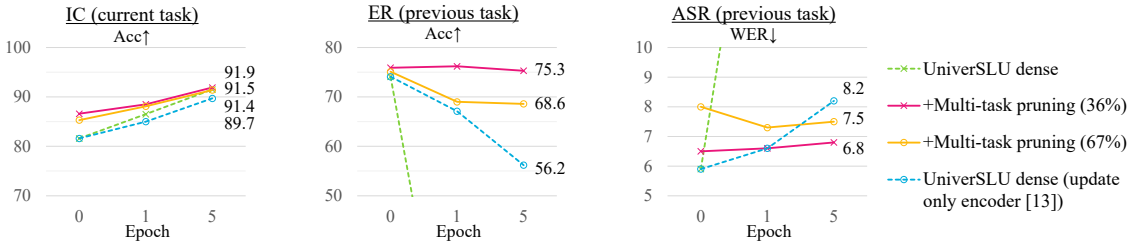
In the first step we identify the task-specific pruning mask, as summarized in Algorithm 1. First, we randomly select a task t at the beginning of the loop (Line 3). Then, we train the network on batches from dataset of task t denoted as $\mathcal{D}_t$ for N_1 iterations (Line 5). After training, pruning is done by setting the pruned position of m_t to 0 (Line 7). In this study, we apply global pruning, where the parameters with the top $p\%$ smallest magnitude across all the layers are pruned. After pruning, the parameters are reset to θ_0 based on LTH [24] (Line 8). We assume iterative pruning, where pruning is repeatedly applied Q times, leading to subnetworks of $1 - (1 - p)^Q\%$ sparsity.

Consequently, we update the parameters based on the identified mask m_t, as in Algorithm 2. Model parameters θ are shared across tasks, which corresponds to multi-task training. Note that each task has its own subnetwork using m_t denoted as $m_t \odot \theta$, so the parameters at subnetwork level are different between tasks. We randomly select task t (Line 3) and train the network on $\mathcal{D}_t$ (Line 5). The training is done for N_2 iterations, where we set a small value ($N_1 \gg N_2$) for the model not to be biased toward the lately trained tasks. In single-task pruning for comparison, Algorithm 2 is performed independently for each task by employing θ_t.

Our multi-task pruning not only brings model compression but also has the ability of continual learning, which has not been investigated in [23]. We assume the case that, when the new data of task t_c become available, we additionally train the model only on the task t_c data. As continual learning in this study, we aim to improve the performance of task t_c, without degrading the performances of other tasks $t \neq t_\mathrm{c}$. For the pruned models, we update only a portion of parameters of a task-specific subnetwork $\{\theta_j | m_{t_\mathrm{c},j} = 1\}$ for continual learning. Other parameters $\{\theta_j | m_{t_\mathrm{c},j} = 0\}$ remain unchanged, some of which is used in a subnetwork for previously trained tasks $t \neq t_\mathrm{c}$. This can play an important role in retaining the knowledge of such tasks. Fi-



(a) *LibriSpeech ASR*



(b) *SLURP IC*

Figure 2: *Continual learning on (a) LibriSpeech ASR and (b) SLURP IC. We additionally trained models on LibriSpeech* 360*h or SLURP real+synthetic data.*

Table 1: *Performances of dense and pruned models. Param(%) indicates percentage of nonzero parameters to perform single task (One) or all tasks (All). Multi-task pruning (our proposal) leads to different subnetworks in shared model, while single-task pruning leads to different models. Task-agnostic pruning means single subnetwork for all task.*

	Param(%) One / All	ER Acc↑	IC Acc↑	ASR WER↓
UniverSLU dense	100 /100	74.1	81.6	5.9
+Task-specific pruning (36%)				
Multi-task	64.1/ 71.0	75.9	86.6	6.5
Single-task	64.1/192.3	77.0	86.7	6.4
+Task-agnostic	64.1/ 64.1	74.2	84.7	6.9
+Task-specific pruning (67%)				
Multi-task	32.9/ 39.8	75.1	85.3	8.0
Single-task	32.9/ 98.6	75.5	85.2	7.9
+Task-agnostic	32.9/ 32.9	74.3	84.3	8.3

nally, our pruning provides some interpretability, by examining the identified structures of task-specifics subnetworks and their overlap across tasks.

4. Experimental evaluations

We conducted experiments using UniverSLU [5] as the multi-task SLU model, obtained by fine-tuning Whisper [6] on SLU tasks. We used Whisper medium of around 760M parameters as the pre-trained model. Our initial investigation is based on three datasets: IEMOCAP [25], SLURP [26], and LibriSpeech [27], which will be extended to 7 tasks in Section 4.2. IEMOCAP was used for ER, which consists of 12 hours of speech, and 4 emotion classes (neutral, happy, sad, and angry) were used in this study. SLURP was used for IC, which consists of 58 hours of real and 43.5 hours of synthetic speech with 69 intent classes. We left the synthetic subset for continual learning. For ASR, we used LibriSpeech the 100h subset in training and test-other set in evaluation. The training of UniverSLU was done on the mixture of the above three datasets, where IEMOCAP was upsampled by 10 times to account for data imbalance. Following the UniverSLU paper [5], we specified which language, task, and dataset to solve by using Whisper-style prompts such as "<EN> <IC> <SLURP>". We conducted the experiments using the ESPnet toolkit [34, 35] and followed its data pre-processing.

First, we trained a dense UniverSLU model for the three SLU tasks. Then, we applied network pruning to find task-specific subnetworks in the dense model as in Algorithm 1 and 2. Algorithm 1 iteratively prunes the network by $p = 20\%$ for $Q = 2$ or 5 times, resulting in approximately 36% and 67% sparsity. We set pruning interval step N_1 equivalent to 3 epochs (average $N_1 = 1300$), where the learning rate was set to 2.0×10^{-4} with 2500 warmup steps. Global pruning was applied to all the layers of both the Whisper encoder and decoder, except for positional embeddings. Algorithm 2 updates model parameters, switching tasks by average $N_2 = 50$ iterations and repeating $R = 200$ times. The learning rate was set to 1.0×10^{-5} with 1500 warmup steps.

Table 1 shows the performances of dense and pruned models with different methods. "Param(%)" denotes what percentage of parameters of the dense model remains, or are not zero. We note what percentages are needed to perform a single task (denoted as "One") and all three tasks (denoted as "All"). We compared multi-task and single-task pruning described in Section 3. Multi-task pruning leads to different pathways m_t within a single model θ, while single-task pruning leads to different models (θ_t with m_t). To perform a single task, the parameter usage is the same. To perform all the tasks, for multi-task pruning, the parameter usage is guaranteed not to exceed 100% and was 71.0% for the 36% pruned model, due to shared parameters across tasks. However, for single-task pruning, the parameter usage was $64.1 \times 3\,(\text{tasks}) = 193\%$, which is not parameter efficient. The issue becomes more significant as the number of tasks increases, as seen in Section 4.2. We also compared these two task-specific pruning with task-agnostic pruning, where pruning is done without distinction of tasks, leading to a single pruned model for all the tasks. In terms of the task performances in Table 1, we found that the accuracy of

Table 2: *Performances of dense and pruned models for 7 tasks.*

	Param(%) One / All	ER Acc↑	IC-SLURP Acc↑	ASR WER↓	IC-FSC Acc↑	IC-SNIPS F1↑	NER SLU F1↑	SCR Acc↑
UniverSLU dense	100 / 100	75.7	85.1	5.9	99.8	93.1	69.4	98.6
+Multi-task pruning	32.9 / 43.7	76.4	84.7	8.1	99.8	94.1	68.4	98.8
+Single-task pruning	32.9 / 230.3	74.7	83.9	8.0	99.7	92.0	68.0	98.7

ER and IC was even improved via pruning. However, we observed performance degradation on ASR, especially for 67% pruning. This would be because ASR is a sequence generation task that requires more parameters than classification tasks ER and IC. We also observed that multi-task pruning achieved competitive performances against single-task pruning, in a parameter efficient way. In case of task-agnostic pruning, all the performances lagged behind those of task-specific pruning. We observed the difference was statistically significant ($p < 0.001$) for IC and ASR, using the Matched Pair test.

4.1. Continual learning

We also investigate how our models work in continual learning settings, assuming the case that new training data for a specific task becomes available. We compared training from the dense UniverSLU model and the multi-task pruned models. We also compared them with updating only the linear layers of encoders in the dense model, which is known as a simple yet effective continual learning method for ASR [13]. We conducted two experiments, using LibriSpeech 360h (ASR) or SLURP synthetic set (IC) as new data. In SLURP experiments, as it is difficult to train the model only on the synthetic speech, we trained them on the mixture of real and synthetic SLURP.

Figure 2a shows the results of the LibriSpeech experiments. As training went on, the ASR performance got improved for all the models, due to additional ASR training data. On the other hand, the ER and IC performances largely deteriorated for the dense model (noted as green lines), known as catastrophic forgetting. The model sometimes outputted ASR results (transcripts), even when it is prompted to perform ER or IC. The problem was mitigated by updating only its encoder (noted as blue lines). For pruned models (noted as yellow and red lines), the performance degradation on ER and IC was smaller compared to the dense model and prior continual learning method. Since only the parameters of the ASR-specific subnetwork are updated during the training, we hypothesize that the remaining parameters can retain the knowledge of other tasks, thus mitigating the issue of catastrophic forgetting. Remarkably, the 36% pruned model even demonstrated an improvement on ER, despite ER not being trained in the continual learning stage. This showcases that shared parameters have the potential to extract features beneficial across tasks.

Figure 2b shows the results of the SLURP experiments. The IC performance was improved by additional data, and the 36% pruned model performed better than the dense model. Similar to LibriSpeech, the performances of ER and ASR were kept better for pruned models, compared to the dense model and prior continual learning method of updating only encoders.

4.2. Extending tasks from 3 to 7

We add IC on Fluent SC (FSC) [36], IC on SNIPS [37], named entity recognition (NER) on SLURP [26], and speech command recognition (SCR) on Google Speech Commands [38] to the training of UniverSLU. SNIPS and FSC consist

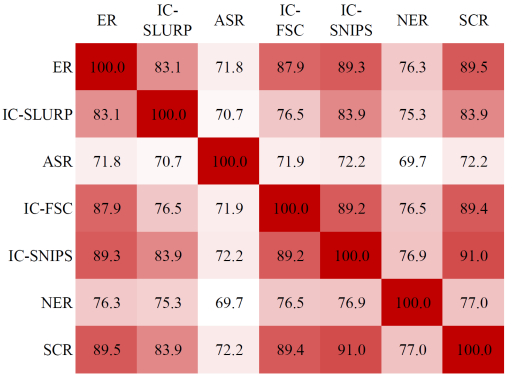


Figure 3: *Parameter overlap ratio between tasks.*

of 1.6K and 30K utterances with 6 and 24 intent classes, respectively. SLURP also has annotations for NER of 55 classes. NER is done by predicting entity tags and corresponding lexical fillers alternately, like "`<entity:date>` `<FILL>` `tomorrow` `<SEP>` ...", similar to [35]. It is evaluated on the SLU-F1 metric introduced in [26]. Google Speech Commands (v0.02) contains 36K utterances for 12 different commands. We regard IC on different datasets as different tasks, as intent labels are different, which results in 7 tasks.

Table 2 shows the performance of the 7-task UniverSLU. Similar to Table 1, pruning was able to reduce the parameter counts with small performance losses or improvements on some tasks, except for ASR. Also, multi-task pruning was parameter efficient to solve multiple tasks and achieved consistently better performances than single-task pruning except for the ASR task.

Finally, Figure 3 analyzes the difference in pruning masks between tasks. We calculated the parameter overlap ratio between subnetworks for task i and j, following [23], as: $\mathrm{Overlap}(m_i, m_j) = \frac{|m_i=1 \cap m_j=1|}{|m_i=1 \cup m_j=1|}$. ASR and NER are sequence generation tasks, while others are classification tasks. This can be the reason why ASR and NER have less overlap with others. Among classification tasks, the overlap ratios are relatively high, where the overlap between IC-SNIPS and SCR is the highest. NER performs generation of entity tags along with lexical fillers, which can make its subnetwork also different from ASR.

5. Conclusions

We have investigated network pruning to obtain task-specific subnetworks within a multi-task SLU model. We conducted experimental evaluations based on UniverSLU model that covers ER, IC, and ASR. We found that subnetworks achieved better performances on ER and IC than the dense network, even with 67% sparsity. In addition to model compression, our approach also has continual learning capabilities. We also found that, with additional ASR training, the ASR performance can be improved without largely degrading the previously trained ER and IC performances. As future work, we plan to extend this study by incorporating structured pruning, as discussed in Section 2.1.

6. References

[1] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[2] K.-W. Chang, W.-C. Tseng, S.-W. Li, and H. yi Lee, "An Exploration of Prompt Tuning on Generative Spoken Language Model for Speech Processing Tasks," in *Interspeech*, 2022.

[3] K.-W. Chang, Y.-K. Wang, H. Shen, I. thing Kang, W.-C. Tseng, S.-W. Li, and H. yi Lee, "SpeechPrompt v2: Prompt tuning for speech classification tasks," *ArXiv*, 2023.

[4] K. Lakhotia, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed, and E. Dupoux, "On generative spoken language modeling from raw audio," *Transactions of the Association for Computational Linguistics*, 2021.

[5] S. Arora, H. Futami, J. weon Jung, Y. Peng, R. Sharma, Y. Kashiwagi, E. Tsunoo, and S. Watanabe, "UniverSLU: Universal spoken language understanding for diverse classification and sequence generation tasks with a single network," *ArXiv*, 2023.

[6] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *ArXiv*, 2022.

[7] J. Wang, Z. Du, Q. Chen, Y. Chu, Z. Gao, Z. Li, K. Hu, X. Zhou, J. Xu, Z. Ma, W. Wang, S. Zheng, C. Zhou, Z. Yan, and S. Zhang, "LauraGPT: Listen, attend, understand, and regenerate audio with gpt," *ArXiv*, 2023.

[8] C. Tang, W. Yu, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. Ma, and C. Zhang, "SALMONN: Towards generic hearing abilities for large language models," *ArXiv*, 2023.

[9] Y. Gong, A. H. Liu, H. Luo, L. Karlinsky, and J. R. Glass, "Joint audio and speech understanding," *ASRU*, 2023.

[10] I. J. Goodfellow, M. Mirza, X. Da, A. C. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgeting in gradient-based neural networks," in *ICLR*, 2014.

[11] H.-J. Chang, H. yi Lee, and L.-S. Lee, "Towards lifelong learning of end-to-end ASR," in *Interspeech*, 2021.

[12] M. Yang, I. Lane, and S. Watanabe, "Online continual learning of end-to-end speech recognition models," in *Interspeech*, 2022.

[13] Y. Takashima, S. Horiguchi, S. Watanabe, P. García, and Y. Kawaguchi, "Updating only encoders prevents catastrophic forgetting of end-to-end asr models," in *Interspeech*, 2022.

[14] A. Diwan, C. feng Yeh, W.-N. Hsu, P. Tomasello, E. Choi, D. F. Harwath, and A. rahman Mohamed, "Continual learning for on-device speech recognition using disentangled conformers," *ICASSP*, 2022.

[15] S. V. Eeckt and H. Van Hamme, "Using adapters to overcome catastrophic forgetting in end-to-end automatic speech recognition," in *ICASSP*, 2023.

[16] U. Cappellazzo, M. Yang, D. Falavigna, and A. Brutti, "Sequence-level knowledge distillation for class-incremental end-to-end spoken language understanding," in *Interspeech*, 2023.

[17] M. Yang, X. Li, U. Cappellazzo, S. Watanabe, and B. Raj, "Evaluating and improving continual learning in spoken language understanding," *CoRR*, 2024.

[18] T. Wu, L. Luo, Y.-F. Li, S. Pan, T.-T. Vu, and G. Haffari, "Continual learning for large language models: A survey," *ArXiv*, vol. abs/2402.01364, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:267406164

[19] 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," in *NeurIPS*, 2021.

[20] S. Ding, T. Chen, and Z. Wang, "Audio lottery: Speech recognition made ultra-lightweight, noise-robust, and transferable," in *ICLR*, 2022.

[21] Y. Peng, K. Kim, F. Wu, P. Sridhar, and S. Watanabe, "Structured pruning of self-supervised pre-trained models for speech recognition and understanding," in *ICASSP*, 2023.

[22] H. Yang, Y. Shangguan, D. Wang, M. Li, P. Chuang, X. Zhang, G. Venkatesh, O. Kalinli, and V. Chandra, "Omni-Sparsity DNN: fast sparsity optimization for on-device streaming E2E ASR via supernet," in *ICASSP*, 2022.

[23] M. Yang, A. Tjandra, C. Liu, D. Zhang, D. Le, and O. Kalinli, "Learning ASR pathways: A sparse multilingual asr model," in *ICASSP*, 2023.

[24] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *ICLR*, 2019.

[25] C. Busso, M. Bulut, C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: interactive emotional dyadic motion capture database," *Lang. Resour. Evaluation*, 2008.

[26] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, "SLURP: a spoken language understanding resource package," in *EMNLP*, 2020.

[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015.

[28] D. W. Blalock, J. J. G. Ortiz, J. Frankle, and J. V. Guttag, "What is the state of neural network pruning?" *CoRR*, 2020.

[29] Z. Li and D. Hoiem, "Learning without forgetting," in *ECCV*, 2016.

[30] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *CoRR*, 2016.

[31] D. Lopez-Paz and M. A. Ranzato, "Gradient episodic memory for continual learning," in *NeurIPS*, 2017.

[32] A. Mallya and S. Lazebnik, "PackNet: adding multiple tasks to a single network by iterative pruning," in *CVPR*, 2018.

[33] S. Golkar, M. Kagan, and K. Cho, "Continual learning via neural pruning," in *NeurIPS*, 2019.

[34] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Interspeech*, 2018.

[35] S. Arora, S. Dalmia, P. Denisov, X. Chang, Y. Ueda, Y. Peng, Y. Zhang, S. Kumar, K. Ganesan, B. Yan, N. Thang Vu, A. W. Black, and S. Watanabe, "ESPnet-SLU: Advancing spoken language understanding through ESPnet," in *ICASSP*, 2022.

[36] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech Model Pre-Training for End-to-End Spoken Language Understanding," in *Interspeech*, 2019.

[37] A. Saade, A. Coucke, A. Caulier, J. Dureau, A. Ball, T. Bluche, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, and M. Primet, "Spoken language understanding on the edge," *EMC2-NIPS*, 2018.

[38] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *CoRR*, 2018.