INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Rehearsal-Free Online Continual Learning for Automatic Speech Recognition

Steven Vander Eeckt, Hugo Van hamme

KU Leuven
Department Electrical Engineering ESAT-PSI, Leuven, Belgium
{steven.vandereeckt, hugo.vanhamme}@esat.kuleuven.be

Abstract

Fine-tuning an Automatic Speech Recognition (ASR) model to new domains results in degradation on original domains, referred to as Catastrophic Forgetting (CF). Continual Learning (CL) attempts to train ASR models without suffering from CF. While in ASR, offline CL is usually considered, online CL is a more realistic but also more challenging scenario where the model, unlike in offline CL, does not know when a task boundary occurs. Rehearsal-based methods, which store previously seen utterances in a memory, are often considered for online CL, in ASR and other research domains. However, recent research has shown that weight averaging is an effective method for offline CL in ASR. Based on this result, we propose, in this paper, a rehearsal-free method applicable for online CL. Our method outperforms all baselines, including rehearsal-based methods, in two experiments. Our method is a next step towards general CL for ASR, which should enable CL in all scenarios with few if any constraints.

Index Terms: automatic speech recognition, online continual learning, catastrophic forgetting, weight averaging

1. Introduction

Catastrophic Forgetting (CF) [1] occurs when Automatic Speech Recognition (ASR) models are extended to new domains (e.g. accents, languages, speakers, topics, etc.) which differ from the original domain the models were trained on. It means that by learning the new domains, the models' performance of the original domain degrades. This severely limits the possibility to build very powerful, diverse and inclusive ASR models, performing well on all dialects, accents, speakers, topics, etc. because the ASR models cannot be properly extended. Learning a new dialect, accent or speaker will result in the model forgetting old dialects, accents or speakers.

Continual Learning (CL) attempts to find strategies to train models without suffering from CF. Within ASR, CL has recently been gaining attention [2, 3, 4, 5, 6, 7, 8, 9]. However, with the exception of [8], the focus of the above research is on offline CL rather than the more challenging online CL.

In offline CL, the model, trained on an initial task, is extended to new tasks. Tasks are represented by training and validation sets which the model has access to until it has learned the given tasks. Moreover, for the model, it is clear when one tasks ends and another starts. In online CL, on other hand, the model receives a stream of batches which it has to process. Once a batch has been learned, access is lost. Moreover, the model does not know whether two consecutive batches belong to the same task or not, i.e. it does not know when a task boundary occurs. Clearly, online CL is a more realistic and generally applicable though also more challenging scenario than offline CL.

To the best of our knowledge, [8] is the only work considering online CL for ASR. In [8], a well-known CL method from computer vision, Gradient Episodic Memory (GEM) [10], is applied to online CL for ASR and referred to as O-GEM (Online GEM). O-GEM, like GEM, is a rehearsal-based method, which means that it attempts to overcome CF by storing previously seen utterances in a small memory. Utterances in this small memory are then later used during training of new batches to prevent forgetting. In computer vision, online CL has received much attention, with most of the proposed methods also being rehearsal-based methods [11, 12, 13, 14, 15], since the gap between rehearsal-based and rehearsal-free methods, which do not use a memory, remains, in particular for online CL, large.

The same could be said about offline CL in ASR [2, 3]. However, recently, [9] found weight averaging (i.e. computing the average of the model before and after being adapted to a new task) to be very effective. Without using a memory, their simple method outperformed rehearsal-based methods; which is significant because storing utterances from previous tasks is not always allowed nor desired. Nevertheless, their method is not applicable to online CL. Based on the simple but very effective method from [9] and inspired by the methods from [14, 15] for computer vision, we propose an online CL method that is rehearsal-free and uses weight averaging. In two experiments, our method outperforms the rehearsal-based method from [8] as well as the rehearsal-based methods applicable to online CL from [3]. We believe that this paper is an important next step towards general CL [13] for ASR, which must enable CL in all scenarios and with few if any constraints, since our method achieves the best performance without requiring a memory and without requiring to know the task boundaries.

2. Model

We consider an encoder-decoder end-to-end ASR model with parameters $\theta \in \mathbb{R}^N$, taking as input speech frames X of size $L_F \times d_i$ with L_F the number of frames. The output tokens of the model are C word pieces. Given ground truth y of L_W outputs tokens, the model's loss consists of a cross-entropy (CE) loss $\mathcal{L}_{\text{dec}}$, computed on the output of the decoder, and a CTC loss $\mathcal{L}_{\text{ctc}}$, computed on the output of the encoder (with $0 \leq c \leq 1$):

$$\mathcal{L}_{\text{ce}}(X, y; \theta) = (1 - c)\mathcal{L}_{\text{dec}}(X, y; \theta) + c\mathcal{L}_{\text{ctc}}(X, y; \theta) \quad (1)$$

3. Online Continual Learning

The objective of continual learning is to learn new tasks without forgetting old ones. Often, it is assumed that tasks boundaries are known and that the data of the tasks remains available until the task has been learned by the model; this is called offline CL. However, for online CL, this is not the case. Access to each

10.21437/Interspeech.2023-788

Table 1: *Results after learning the stream of batches from Fig. 1a. All WERs (expressed in percentages) are evaluated on the final model. † indicates that the method's hyper-parameters were optimized on the test experiment. Best AWER result is in bold.*

			WER per task						
Model	M	(τ, λ, τ_2)	$\mathcal{T}_0$–US	$\mathcal{T}_1$–ENG	$\mathcal{T}_2$–AUS	$\mathcal{T}_3$–IND	$\mathcal{T}_4$–SCO	$\mathcal{T}_5$–IRE	**AWER**
Initial model θ_0			17.3	13.9	15.4	21.4	15.2	11.0	15.72
FT			19.4	15.0	15.2	26.1	15.5	11.5	17.12
UOE			18.3	12.6	13.1	21.6	14.2	11.0	15.15
EWC†			18.3	13.4	14.1	22.4	14.7	11.1	15.66
ER†	$0.5k$		18.1	13.2	13.9	22.1	14.7	11.1	15.50
ER†	$2.0k$		17.9	12.9	13.6	21.5	14.1	10.8	15.13
O-GEM	$2.0k$		19.0	14.1	14.4	25.2	14.9	11.3	16.48
AOS		$(1, 0.1, 1)$	17.1	12.7	14.0	21.4	14.3	10.7	15.03
AOS†		$(2, 0.1, 1)$	17.3	12.5	13.6	21.5	14.2	10.7	**14.96**

sample or batch is lost once it has been seen by the model and the task boundaries are unknown.

Formally, a model θ_0 has been trained on an initial task $\mathcal{T}_0$ with data $\mathcal{D}_0$, containing D_0 samples. Next, the model receives a (non-i.i.d.) stream of batches $\mathcal{B}_i$ $(i > 0)$ of size B, which it processes batch after batch. When learning $\mathcal{B}_i$, access to $\mathcal{B}_{i-1}$ is lost, except possibly through a memory $\mathcal{M}_{i-1}$ of fixed size M. Batches belong to a certain task $\mathcal{T}_{t_i}$, but this is not known by the model. The model should learn all batches well while retaining the knowledge from old batches and tasks.

4. Method

Our method, which we call AOS (**A**veraging for **O**nline CL of A**S**R), consists of two parts: averaging (Sec. 4.1) and regularization (Sec. 4.2). An overview is given in Algorithm 1.

4.1. Online Averaging

Inspired by the effectiveness of weight averaging for CL in ASR [9] and the methods of [14, 15], we consider what we refer to as 'online averaging' for our online CL method. In [9], the model is first adapted to a new task, after which the average was computed between the model before and after adaptation. The weight of the adapted model is $\eta = 1/t$ with t the number of seen tasks. However, in [9], task boundaries are assumed to be known, which is not the case here. Therefore, we consider 'online averaging', i.e. we average the 'old' and 'adapted' model after each batch. In other words, if θ_i is the 'final' model at batch $\mathcal{B}_i$, and $\tilde{\theta}_{i+1}$ is the model adapted to a batch $\mathcal{B}_{i+1}$, then the final model after batch $\mathcal{B}_{i+1}$ is:

$$\theta_{i+1} = (1 - \eta_{i+1})\theta_i + \eta_{i+1}\tilde{\theta}_{i+1} \tag{2}$$

Two questions remain here.

The adapted model. What is $\tilde{\theta}_{i+1}$? If $\tilde{\theta}_{i+1}$ is θ_i trained on a new batch $\mathcal{B}_{i+1}$, then θ_{i+1}, instead of through Eq. 2, could be obtained by learning the new batch with a η_{i+1} times smaller learning rate than $\tilde{\theta}_{i+1}$, i.e. our method would just be fine-tuning on the new batch with a η_{i+1} times smaller learning rate. We do not expect this to work well. Therefore, we keep two models: one, θ_{i+1}, which we call the final model; and another, $\tilde{\theta}_{i+1}$, the adapted model, which starts from θ_0 but is then adapted to the new batches and thus from this point on deviates from θ_i. It is very likely that the adapted model suffers from forgetting; at inference time, only the final model should thus be considered. The adapted model is only to aid the final model by transferring new information to it. This resembles the methods from [14, 15], which focus on computer vision and were inspired by the Complementary Learning Systems theory [16].

Weighted average. What is the value of η_{i+1}? Most similar to $\eta = 1/t$ from [9] would be $\eta_{i+1} = B/(D_i+B)$, where D_i is the amount of data the model has seen so far and B is the current batch size. However, we make a number of improvements:

1. The model receives as input utterances consisting of a number of frames, L_F. To allow longer utterances to have an higher impact, we consider F, the number of frames in a batch, rather than the batch size B. F_i is then the number of frames seen after processing batch i, and F_0 is the total number of frames of initial task $\mathcal{T}_0$.
2. For the decoder of the model, the length of an utterance is related to its number of output tokens L_W. Therefore, for the decoder, we consider W, the number of output tokens in a batch, rather than the number of frames F; similar to F_i and F_0, W_i and W_0 are then the total number of output tokens after processing batch i and the number of output tokens in initial task $\mathcal{T}_0$, respectively. This means that the encoder and decoder have separate values $\eta_{\text{enc},i+1}$ and $\eta_{\text{dec},i+1}$ to compute the average in Eq. 2 after processing batch $i + 1$.
3. If the original task was a very large one, i.e. if F_0 (W_0) is very large, then $\eta_{\text{enc},i+1}$ ($\eta_{\text{dec},i+1}$) will be very small and it will take a long time for the encoder (decoder) of the model to learn new information. Therefore, we introduce $\tau \geq 1$ to increase the plasticity of the model:

$$\eta_{\text{enc},i+1} = \frac{\tau \cdot F}{F_i + \tau \cdot F} \tag{3}$$

   and the same for $\eta_{\text{dec},i+1}$, where we then consider W and W_i.
4. For monolingual experiments, the encoder is more prone to forgetting than the decoder [4]. On the other hand, [6] shows that freezing the decoder might overcome forgetting. Therefore, we consider τ_2 for $\eta_{\text{dec},i+1}$:

$$\eta_{\text{dec},i+1} = \frac{\tau_2 \cdot W}{W_i + \tau_2 \cdot W} \tag{4}$$

   With (τ, τ_2), the decoder might be updated more conservatively ($\tau_2 \leq \tau$) or progressively ($\tau_2 \geq \tau$) than the encoder.

In summary, the final model θ_{i+1} is obtained after averaging (Eq. 2) with the adapted model $\tilde{\theta}_{i+1}$, which is trained on the stream of batches, using the weight $\eta_{\text{enc},i+1}$ from Eq. 3 for the encoder and $\eta_{\text{dec},i+1}$ from Eq. 4 for the decoder.

Algorithm 1 AOS (**A**veraging for **O**nline CL of **ASR**)

Given: initial model θ_0 trained on data $\mathcal{D}_0$ with D_0 utterances, F_0 frames and W_0 output tokens.
Choose: α, c (for model); τ, λ, τ_2 (for AOS)
Set: $i \leftarrow 0$, $\tilde{\theta}_0 \leftarrow \theta_0$
 # online continual learning:
for each batch $(X, y) \in \mathcal{B}_{i+1}$ **do**
    # compute the loss on new batch for adapted model
    $\mathcal{L} \leftarrow \mathcal{L}(X, y; \tilde{\theta}_i)$ ▷ See Eq. 7
    # update adapted model with SGD
    $\tilde{\theta}_{i+1} \leftarrow \tilde{\theta}_i - \alpha \nabla \mathcal{L}$
    # F: number of frames in X
    $\eta_{\text{enc},i+1} \leftarrow \tau F / (F_i + \tau F)$ ▷ See Eq. 3
    # W: number of output tokens in y
    $\eta_{\text{dec},i+1} \leftarrow \tau_2 W / (W_i + \tau_2 W)$ ▷ See Eq. 4
    # use $\eta_{dec,i+1}$ for decoder parameters, else use $\eta_{enc,i+1}$
    $\eta_{i+1} \leftarrow \eta_{\text{dec},i+1}$ if decoder layer else $\eta_{\text{enc},i+1}$
    # update the final model
    $\theta_{i+1} \leftarrow (1 - \eta_{i+1})\theta_i + \eta_{i+1}\tilde{\theta}_{i+1}$ ▷ See Eq. 2
    # update seen frames F_i and seen outputs W_i
    $(F_{i+1},\ W_{i+1}) \leftarrow (F_i + F,\ W_i + W)$
    $i \leftarrow i + 1$
end for

4.2. Regularization

We apply regularization to the adapted model to improve the performance of the final model. We consider knowledge distillation (KD) [17] as in Learning without Forgetting (LWF) [18], a popular CL method that uses the data of the current batch to distill knowledge from the old to the current model. With $\hat{s}^i_{\text{ctc},kc}$ and $s^i_{\text{ctc},kc}$ the CTC softmax output of the cth word piece at the kth frame of, respectively, the final and adapted model after batch i, the KD loss for the CTC output becomes:

$$\mathcal{L}_{\text{ctc, kd}}(X; \theta) = \sum_{k=1}^{L_F} \sum_{c=1}^{C} \hat{s}^i_{\text{ctc},kc} \log s^i_{\text{ctc},kc} \quad (5)$$

The KD loss for the decoder, $\mathcal{L}_{\text{dec, kd}}(X, y; \theta)$, is computed similarly, by replacing, in Eq. 5, the CTC softmax outputs by the decoder softmax outputs, with the outer sum then summing over all L_W outputs. Since the decoder is autoregressive, its KD loss depends on the ground truth y. Overall, the KD loss becomes:

$$\mathcal{L}_{\text{kd}}(X, y; \theta) = (1 - c)\mathcal{L}_{\text{dec, kd}}(X, y; \theta) + c\mathcal{L}_{\text{ctc, kd}}(X; \theta) \quad (6)$$

With λ the regularization weight, the above KD regularization loss is then added to the model's loss from Eq. 1 as follows:

$$\mathcal{L}(X, y; \theta) = (1 - \lambda)\mathcal{L}_{\text{ce}}(X, y; \theta) + \lambda\mathcal{L}_{\text{kd}}(X, y; \theta) \quad (7)$$

The KD loss transfers knowledge from the final to the adapted model and, as such, regularizes the training of the latter, which then improves the performance of the former.

5. Experiments

All experiments are done in ESPnet2 [19]. For all information regarding the experiments, we refer to our Github repository [1].

Data. We consider English data of Common Voice (CV) [20], split into six accents: United States (US), England (ENG), Australia (AUS), India (IND), Scotland (SCO), Ireland (IRE). The initial model θ_0 is trained on an initial task $\mathcal{T}_0$; the batches of the remaining five tasks are sorted by task and by speaker and as such presented to the model. This makes the experiment more challenging, since the model is susceptible to forgetting both across tasks (accents) and within tasks (speakers). We consider two sequences of the tasks. Both take US as the initial task $\mathcal{T}_0$, since US is by far the largest task ($350k$ utterances). We consider this to be the most realistic scenario in practice. For the five remaining tasks ($262k$ utterances), we consider the two sequences as shown in Fig. 1: ENG→AUS→IND→SCO→IRE and IRE→IND→AUS→ENG→SCO.

Model. The model ($47M$ parameters) consists of 12 Conformer [21] encoders and 6 Transformer [22] decoders of dimension 2048, with 4 attention heads with dimension 256. The output are $C = 5000$ word pieces generated by Sentence Piece [23] on $\mathcal{T}_0$. The weight of the CTC loss is $c = 0.3$. The model is trained on initial task $\mathcal{T}_0$ for 80 epochs. Afterwards, it learns the stream of batches $\mathcal{B}_i$ ($i > 0$), seeing each batch only once. The batch size is 32, but since each batch only contains one speaker, in practice the average batch size is 22. The model is updated with the SGD optimizer with a learning rate of 0.01.

Baselines. We consider the following baselines:

- *Fine-Tuning (FT)*: adaptation without regularization. FT is considered the worst case baseline and will suffer from CF.
- *Experience Replay (ER)* [11]: trains jointly on the new batch and a batch sampled from memory. We consider the implementation of [3] with regularization weight.
- *Online Gradient Episodic Memory (O-GEM)* [8]: online implementation of GEM [10], which updates the gradient before the SGD update to prevent interference with previous tasks. We sample randomly from the memory.
- *Update Only Encoders (UOE)* [6]: proposes to only update the encoders (without layer normalization) to overcome CF.
- *Elastic Weight Consolidation (EWC)* [24]: computes for all parameters 'importance weights', used in a weighted L2 regularization loss. We consider the online version of [25].

For the rehearsal-based methods (ER and O-GEM), we consider reservoir sampling [26] to fill the memory of size $M = 2k$.

Metrics. We report the WER per task, as well as average WER (AWER), averaged over all seen tasks (accents).

Hyper-parameters. For the hyper-parameters, we run hyper-parameter searches on a 'test experiment', by adapting the model trained on the initial task $\mathcal{T}_0$ = US to some small accents from CV not present in one of the six tasks. These accents account for only $13k$ utterances, so approximately 5% of the 'real' experiment. Next, we consider the AWER between the validation sets of US and the new task to find the optimal hyper-parameter value. Since the test experiment contains only a small number of utterances, we put additional focus on 'not forgetting' by giving the WER on US in the computation of AWER a higher weight than the WER on the new task (2 vs. 1).

6. Results

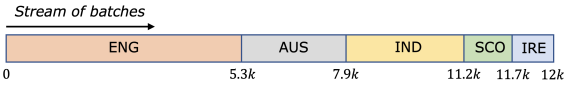
Tables 1 and 2 show the results of the experiments for the two sequences. The performance of FT and its forgetting illustrate that CF remains a serious issue, even if the new domains (tasks) are not so dissimilar to the original one (i.e. different accents). In addition, from Tab. 1, we observe the following:

- Our method, AOS, outperforms all baselines, even with the default (i.e. non-optimized) values for its hyper-parameters. These baselines include ER and O-GEM, which have access

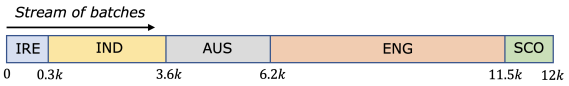
[1] https://github.com/StevenVdEeckt/online-cl-for-asr

Table 2: *Results after learning the stream of batches from Fig. 1b. All WERs (expressed in percentages) are evaluated on the final model. † indicates that the method's hyper-parameters were optimized on the test experiment. Best AWER result is in bold.*

			WER per task						
Model	M	(τ, λ, τ_2)	$\mathcal{T}_0$–US	$\mathcal{T}_1$–IRE	$\mathcal{T}_2$–IND	$\mathcal{T}_3$–AUS	$\mathcal{T}_4$–ENG	$\mathcal{T}_5$–SCO	**AWER**
Initial model θ_0			17.3	11.0	21.4	15.4	13.9	15.2	15.72
FT			19.1	11.5	25.3	14.2	13.1	14.5	16.27
UOE			18.7	11.7	24.2	13.0	12.2	14.3	15.68
EWC†			17.9	11.1	22.9	14.1	12.6	15.0	15.60
ER†	$2.0k$		17.8	10.8	21.4	13.9	12.5	14.2	15.11
O-GEM	$2.0k$		19.0	11.5	25.3	14.0	13.0	14.3	16.18
AOS		$(1, 0.1, 1)$	17.2	10.7	21.2	14.3	12.9	14.4	15.11
AOS†		$(2, 0.1, 1)$	17.5	10.7	21.5	13.7	12.6	14.3	**15.03**



(a) *Sequence 1*



(b) *Sequence 2*

Figure 1: *The stream of batches as presented to the model. The axis shows the batch number (average batch size is 22). The batches are sorted by speaker and split into five accents/tasks, as shown. The task boundaries are not known by the model.*

to a memory of $M = 2k$ utterances. This is a relatively large memory, given that the $262k$ utterances from the stream of batches are only seen once. If a smaller memory has to be used (e.g. $M = 0.5k$), the performance of ER deteriorates and the gap with our (rehearsal-free) method becomes larger. UOE, which freezes the decoder and norm layers of the initial model θ_0, is the second best baseline. It is also rehearsal-free and works well, yet is still outperformed by our method.

- In particular the performance of the methods on $\mathcal{T}_0$–US is interesting to compare, since this is the initial task the model was trained on. We can see that none of the baselines come close to the zero forgetting that our method achieves (by comparing it to *Initial model* θ_0), and the gap between our method and the best rehearsal-free baseline, UOE, in this regard becomes large, since UOE suffers from CF. Our method with the default setting even achieves positive backward transfer, i.e. by learning new tasks it improves on old ones.
- Even in the default setting, AOS outperforms all baselines. Nevertheless, when optimizing (τ, λ, τ_2) on the test experiment, we obtain even better performance, though the difference is small. With the optimal hyper-parameters, the encoder is averaged more progressively than the decoder ($\tau > \tau_2$). As a result, there is no positive backward transfer on $\mathcal{T}_0$–US, however, on the new tasks, the model performs better thanks to increased plasticity ($\tau > 1$).

Similar observations apply to the second sequence (Tab. 2):

- Our method outperforms all baselines, with the exception of ER for the default setting of our method; they achieve the same performance. With the optimal hyper-parameters, AOS outperforms all baselines, including ER. Note also that here UOE is much less effective than in Tab. 1. Consequently, the gap between our method and the rehearsal-free baselines (UOE and EWC) is wide; only our method is able to compete with the rehearsal-based baselines.
- Our method achieves the best performance on the initial task $\mathcal{T}_0$–US. Again, the default setting of our method achieves positive backward transfer, while the optimal setting achieves slight forgetting, though still better than the other methods.
- The optimal setting again outperforms the default setting, though the difference is small and it comes at the cost of some forgetting for the initial task. However, increased plasticity ($\tau > 1$) enables the model to better learn the new tasks for the optimal setting of our method.

Overall, it can be seen that our method is highly effective, surpassing, even in its default setting, all baselines including those with large memory and this in both sequences. In addition, our method achieves zero forgetting unlike the baselines.

7. Conclusions

We propose a very effective and simple method for online CL for end-to-end ASR that involves two models: an adapted model and final model. The adapted model is trained on new batches with regularization, and its parameters are then averaged with those of the final model to update the final model. The weight of the averaging is determined by the number and length of utterances in the new batch compared to those previously seen. The averaging transfers knowledge from the adapted model to the final model, allowing it to learn new tasks without forgetting old ones. This is illustrated by the experiments, in which our method, even in its default setting, outperforms all baselines, including those with large memory. Our method is rehearsal-free, making it simpler and more efficient than other approaches.

By overcoming the need for task boundaries and/or a memory, our method takes a step towards a more general CL method for ASR that can work in many different scenarios (when storing utterances is not allowed and/or task boundaries unknown). In future work, we aim to further extend our method in two ways, by allowing it to: (1) learn new batches in an unsupervised way; (2) introduce new word pieces to the vocabulary if needed. While this paper is an important next step towards general CL for ASR, these two objectives will be our focus in the future to achieve a truly general CL method for ASR.

8. Acknowledgments

Research supported by Research Foundation Flanders (FWO) under grant S004923N of the SBO programme.

9. References

[1] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," ser. Psychology of Learning and Motivation, 1989, vol. 24, pp. 109–165. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079742108605368

[2] H.-J. Chang, H. yi Lee, and L. shan Lee, "Towards Lifelong Learning of End-to-End ASR," in *Proc. Interspeech 2021*, 2021, pp. 2551–2555.

[3] S. Vander Eeckt and H. Van hamme, "Continual learning for monolingual end-to-end automatic speech recognition," *Proceedings EUSIPCO 2022*, 2022. [Online]. Available: $$Uhttps://lirias.kuleuven.be/retrieve/666478$$D4698_preprint.pdf[AvailableforKULeuvenusers-Embargoeduntil2022-09-02]

[4] S. V. Eeckt and H. Van Hamme, "Using adapters to overcome catastrophic forgetting in end-to-end automatic speech recognition," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[5] M. Sustek, S. Sadhu, and H. Hermansky, "Dealing with Unknowns in Continual Learning for End-to-end Automatic Speech Recognition," in *Proc. Interspeech 2022*, 2022, pp. 1046–1050.

[6] Y. Takashima, S. Horiguchi, S. Watanabe, L. P. García-Perera, and Y. Kawaguchi, "Updating only encoders prevents catastrophic forgetting of end-to-end ASR models," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, H. Ko and J. H. L. Hansen, Eds. ISCA, 2022, pp. 2218–2222. [Online]. Available: https://doi.org/10.21437/Interspeech.2022-11282

[7] A. Diwan, C.-F. Yeh, W.-N. Hsu, P. Tomasello, E. Choi, D. Harwath, and A. Mohamed, "Continual learning for on-device speech recognition using disentangled conformers," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

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

[9] S. Vander Eeckt and H. Van Hamme, "Weight averaging: A simple yet effective method to overcome catastrophic forgetting in automatic speech recognition," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[10] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, 2017, p. 6470–6479.

[11] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Paper.pdf

[12] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 11 849–11 860. [Online]. Available: http://papers.nips.cc/paper/9357-online-continual-learning-with-maximal-interfered-retrieval.pdf

[13] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. CALDERARA, "Dark experience for general continual learning: a strong, simple baseline," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 15 920–15 930. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/b704ea2c39778f07c617f6b7ce480e9e-Paper.pdf

[14] E. Arani, F. Sarfraz, and B. Zonooz, "Learning fast, learning slow: A general continual learning method based on complementary learning system," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=uxxFrDwrE7Y

[15] F. Sarfraz, E. Arani, and B. Zonooz, "Synergy between synaptic consolidation and experience replay for general continual learning," in *Proceedings of The 1st Conference on Lifelong Learning Agents*, ser. Proceedings of Machine Learning Research, S. Chandar, R. Pascanu, and D. Precup, Eds., vol. 199. PMLR, 22–24 Aug 2022, pp. 920–936. [Online]. Available: https://proceedings.mlr.press/v199/sarfraz22a.html

[16] D. Kumaran, D. Hassabis, and J. L. McClelland, "What learning systems do intelligent agents need? complementary learning systems theory updated," *Trends in Cognitive Sciences*, vol. 20, no. 7, pp. 512–534, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364661316300432

[17] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *NeurIPS*, vol. abs/1503.02531, 2014. [Online]. Available: http://arxiv.org/abs/1503.02531

[18] Z. Li and D. Hoiem, "Learning without forgetting," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer International Publishing, 2016, pp. 614–629.

[19] S. Watanabe *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[20] R. Ardila *et al.*, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, 2020, pp. 4211–4215.

[21] A. Gulati, J. Qin, C.-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*, 2020, pp. 5036–5040.

[22] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[23] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2018, pp. 66–71. [Online]. Available: https://aclanthology.org/D18-2012

[24] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017. [Online]. Available: https://www.pnas.org/content/114/13/3521

[25] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress and compress: A scalable framework for continual learning," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 4528–4537. [Online]. Available: https://proceedings.mlr.press/v80/schwarz18a.html

[26] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, p. 37–57, mar 1985. [Online]. Available: https://doi.org/10.1145/3147.3165