



Towards Lifelong Learning of End-to-end ASR

Heng-Jui Chang, Hung-yi Lee, Lin-shan Lee

School of Electrical Engineering and Computer Science, National Taiwan University, Taiwan

{b06901020, hungyilee}@ntu.edu.tw, lslee@gate.sinica.edu.tw

Abstract

Automatic speech recognition (ASR) technologies today are primarily optimized for given datasets; thus, any changes in the application environment (e.g., acoustic conditions or topic domains) may inevitably degrade the performance. We can collect new data describing the new environment and fine-tune the system, but this naturally leads to higher error rates for the earlier datasets, referred to as catastrophic forgetting. The concept of lifelong learning (LLL) aiming to enable a machine to sequentially learn new tasks from new datasets describing the changing real world without forgetting the previously learned knowledge is thus brought to attention. This paper reports, to our knowledge, the first effort to extensively consider and analyze the use of various approaches of LLL in end-to-end (E2E) ASR, including proposing novel methods in saving data for past domains to mitigate the catastrophic forgetting problem. An overall relative reduction of 28.7% in WER was achieved compared to the fine-tuning baseline when sequentially learning on three very different benchmark corpora. This can be the first step toward the highly desired ASR technologies capable of synchronizing with the continuously changing real world.

Index Terms: lifelong learning, continual learning, end-to-end automatic speech recognition

1. Introduction

The real world is changing and evolving from time to time, and therefore machines naturally need to update and adapt to the new data they receive. However, when a trained deep neural network was adapted to a new dataset with a different distribution, it often loses the knowledge previously acquired and performs the previous task worse than before. This phenomenon is called *catastrophic forgetting* [1]. Under this scenario, people try to re-train the models from scratch with both the past and the new data jointly, sometimes referred to as *multitask learning*. For various reasons, including privacy issues and the limited storage capacity, the earlier data are unlikely to be kept forever. Therefore, *lifelong learning* (LLL) or *continual learning* [2], aiming for training a single model to perform a stream of tasks without forgetting those learned earlier, not relying on keeping all training data from the beginning, becomes a necessary goal for the continuously changing real world.

In general, LLL approaches can be categorized into three types. Regularization-based methods aim to consolidate essential parameters in a model by adding regularization terms in the loss function [3–7]. Architecture-based methods try to assign some model capacity for each task or expand the model size to handle additional tasks [8–10]. Data-based methods then try to save or generate some samples from the past tasks to prevent catastrophic forgetting [11–15]. Studies of LLL have been reported more on computer vision [3, 5–7, 9–12, 14–19] and reinforcement learning [3, 4, 8, 9, 20], yet much less on automatic speech recognition (ASR) tasks [21–24].

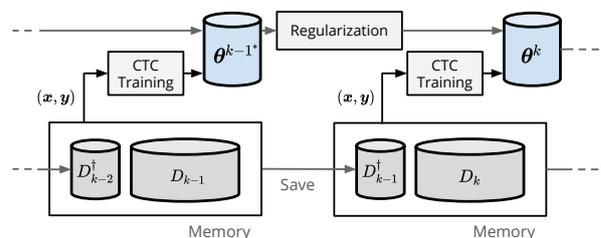


Figure 1: The training framework for LLL for E2E ASR. The k^{th} model (θ^k) is being trained on corpus D_k and can reuse some limited past samples stored in D_{k-1}^\dagger . Also, the model trained in the previous stage (θ^{k-1*}) can be used for regularization.

ASR technologies are very successful globally, and end-to-end (E2E) ASR approaches [25–28] are very powerful in recent years, but with performance inevitably degraded in almost all cases over data disparate from training sets, e.g., in acoustic conditions or topic domains. Various domain adaptation approaches for ASR were shown successful [29–31] in new domains, although inevitably suffering from serious catastrophic forgetting [21–23]. As a result, ASR technologies today remain unable to evolve with the changing real world.

To our knowledge, this is the first paper extensively studying the various concepts of LLL applied to E2E ASR. Sadhu and Hermansky [22] used model expansion for LLL on HMM-DNN ASR. Houston and Kirchhoff [23] used regularization methods for multi-dialect acoustic models. However, E2E ASR discussed here jointly considers acoustic and language modeling in a single network and is thus different and challenging. We compare and analyze regularization- and data-based methods and found the latter very effective, including proposing data selection approaches based on either perplexity or utterance duration. Evaluation of CTC [25] ASR on WSJ [32], LibriSpeech [33], and Switchboard [34] showed an overall relative WER reduction of 28.7% compared to the fine-tuning baseline.

2. Methods

2.1. General Training Framework

Consider a training framework as in Fig. 1 with K training corpora D_1 to D_K from different domains. The E2E ASR (CTC [25] in this paper) is first trained with D_1 , with parameters obtained denoted as θ^{1*} , where “*” indicates the best parameter set. The model is then trained in each stage (k^{th} stage in the right part of Fig. 1) on one corpus (D_k) at a time, and is allowed to reuse some samples from the previous corpora (D_{k-1}^\dagger) stored in a memory with a fixed capacity. The target of LLL is to preserve high recognition accuracy for previous domains, or the last CTC model (θ^{K*}) perform well on all domains D_1 to D_K . All training data are transcribed audio-text pairs (x, y) . The i^{th} parameter of the CTC model trained on D_k is θ_i^k .

2.2. Regularization-based Methods

Elastic Weight Consolidation (EWC). We adopt the EWC [3] and online EWC algorithms [4] previously proposed. A regularization term is used to constrain parameters to stay close to those for previous tasks. The loss function can be written as

$$\mathcal{L}(\theta^k) = \mathcal{L}_{\text{CTC}}(\theta^k) + \frac{\lambda}{2} \sum_i \Omega_i^k (\theta_i^k - \theta_i^{k-1*})^2, \quad (1)$$

where \mathcal{L}_{CTC} is the CTC loss, and Ω_i^k for the importance of each parameter is the diagonal of the Fisher information matrix.

Synaptic Intelligence (SI). We also adopt SI [5], which is similar to EWC, but the importance measure Ω_i^k in Eq. (1) is estimated iteratively by its contribution to decreasing the loss,

$$\Omega_i^k = \Omega_i^{k-1} + \frac{\omega_i^{k-1}}{(\theta_i^{k-1*} - \theta_i^{k-2*})^2 + \xi}, \quad (2)$$

where ω_i^{k-1} is obtained with the gradient of CTC loss and ξ is a small constant to stabilize training.

Knowledge Distillation (KD). Instead of limiting the parameters directly as done previously, KD minimizes the KL divergence between the output distributions of the current model (θ^k) and the previous model (θ^{k-1*}) [12, 21, 24, 35, 36]. The KD loss function can be written as

$$\mathcal{L}_{\text{KD}} = \text{KL} [y'_{k-1} || y'_k], \quad (3)$$

where y'_{k-1} and y'_k are the output probability sequences for the previous and the current models θ^{k-1*} and θ^k , but with the logits z scaled by a temperature $T > 0$ as $y' = \text{softmax}(z/T)$. \mathcal{L}_{KD} then replaces the second term of Eq. (1).

These regularization-based methods require storing the previous model’s weights or Ω_i^k , while the parameters in a CTC model usually occupy a large space (10M+ parameters). The data-based methods below store samples from previous datasets, but the required capacity is unnecessarily larger.

2.3. Data-based Methods

Gradient Episodic Memory (GEM). Here we store samples from the past to calculate the gradients [11]. If the current gradient g increases loss on any of the past domains, it is projected to the gradient \tilde{g} with the minimum L2 distance to g , or

$$\begin{aligned} & \underset{\tilde{g}}{\text{minimize}} && \|g - \tilde{g}\|_2^2 \\ & \text{subject to} && \langle \tilde{g}, g_{k-1} \rangle \geq 0 \end{aligned} \quad (4)$$

where g and g_{k-1} are respectively the gradients of the CTC outputs over the current dataset D_k and the memory D_{k-1}^\dagger , and $\langle \cdot, \cdot \rangle$ is the inner product with positive values implying similar directions. In this paper, we constrain the capacity of D_{k-1}^\dagger to a fixed size and balance each corpus to having the same data size, so with more new tasks, some previous samples have to be dropped. Conventionally the data preserved for GEM are sampled randomly from the previous datasets. For the preserved data to generate gradients representing better directions for the whole dataset, we propose two data selection methods to find samples better indicating previous data distributions.

Minimum Perplexity (PP). Since topic domains vary among datasets, we propose to train an LM (RNN-LM or n-gram-LM) for each dataset D_k , based on which utterances with minimum perplexity are saved in the memory, assuming they better represent the linguistic property of the corpus.

Median Length (Len). We noted the averaged utterance length in each dataset varies, longer in books while shorter in daily and spontaneous conversations. Since longer and shorter utterances have slightly different acoustic properties, we propose to preserve samples with lengths close to the median.

3. Experiments

3.1. Datasets

We chose three corpora with different acoustic and topic domains to form a sequence of tasks for the ASR models to learn.

Wall Street Journal (WSJ) [32]. We used the si-284 set as one of the training sets and the eval92 set for evaluation.

LibriSpeech (LS) [33]. We used the 100-hour clean set as one of the training sets and the clean testing set for evaluation.

Switchboard (SWB) [34]. We chose the 300-hour LDC97S62 subset as one of the training sets and the Hub5-2000 subset for evaluation, more spontaneous and noisy compared to WSJ and LS. We followed the Kaldi [37] "s5c" recipe to process SWB.

3.2. Model

In this paper, the CTC model [25] was considered for E2E ASR. During evaluation, the transcription of a given utterance was either directly decoded from the CTC output distribution or with beam decoding with an additional LM. We considered the LLL for CTC only but not for LM since text data are easier to collect than transcribed speech. The training targets of CTC and LMs were both BPE subwords [38] of size 256 trained on the 800M-word LM corpus from LibriSpeech.

CTC Model. The CTC model [25] was composed of a 2-layer CNN for downsampling and a 5-layer BLSTM of 512 units per direction. We extracted 80-dimensional Mel filterbank features with delta, delta-delta and normalization. The sample rate of SWB is 8kHz, lower than the other two corpora; we thus up-sampled all data to 16kHz. SpecAugment [39] and speed perturbation [40] were performed in all experiments.

Language Model. The RNN-LM was a 2-layer LSTM of 512 units, trained with all text data from the three datasets.

Single Task Results. Row (a) of Table 1 lists the results as references for our CTC trained and tested on every single task without (Sec. (I)) and with (Sec. (II)) LM. These were the best results after trying several models and output units to balance the performance for the three tasks. Although different from the state-of-the-art, these WERs showed that our models worked properly with all three datasets.

3.3. Lifelong Learning with CTC

We then trained the CTC model in the order of WSJ-LS-SWB (the dataset size increased and the data got more spontaneous and noisy incrementally) and then tested on the three individual corpora. The results are listed in Table 1.

3.3.1. Fine-tuning Baseline and Multitask Upper Bound

We set a baseline in row (b) of Table 1 by fine-tuning the models successively stage by stage on the three corpora without doing anything more. Now we focus on CTC model without LM (Sec. (I)). We found results on SWB very similar (column (iii), rows (a) v.s. (b)), showing the previous two tasks provided no gain for the new domain. Results on WSJ and LS were seriously degraded after training with different tasks (columns (i)(ii), rows (b) v.s. (a)), which is an evidence of catastrophic forgetting.

Table 1: WERs(%) of the CTC model without (Sec. (I)) and with (Sec. (II)) LM rescoring trained with different LLL approaches under the training order of WSJ-LS-SWB and tested on the three corpora in columns (i)(ii)(iii) and (vi)(vii)(viii). Columns (iv)(ix) (AVG) and (v)(x) (WERR) are respectively the average indicating the performance level and the relative WER reduction compared with fine-tuning (row (b)). The single corpus baseline, fine-tune baseline, and the multitask upper bound are respectively in rows (a), (b) and (i).

Method	(I) CTC					(II) CTC + RNN-LM				
	(i) WSJ	(ii) LS	(iii) SWB	(iv) AVG	(v) WERR	(vi) WSJ	(vii) LS	(viii) SWB	(ix) AVG	(x) WERR
Baseline										
(a) Single	14.2	13.7	28.7	18.9	—	11.8	10.8	23.0	15.2	—
(b) Fine-tune	25.1	38.8	28.8	30.9	—	18.9	31.4	23.7	24.7	—
Regularization-based										
(c) EWC	25.1	39.3	30.2	31.6	−2.3%	19.1	31.8	24.7	25.2	−2.0%
(d) SI	21.9	32.0	35.7	29.9	3.2%	15.8	23.5	28.6	22.6	8.5%
(e) KD	22.7	33.1	29.4	28.4	8.1%	16.7	25.4	24.3	22.2	10.1%
Data-based										
(f) GEM	23.6	28.2	30.4	27.4	11.3%	17.1	21.9	24.8	21.3	13.8%
(g) GEM + PP	22.8	27.7	30.3	26.9	13.3%	17.1	21.4	24.8	21.1	14.6%
(h) GEM + Len	22.4	27.8	30.1	26.8	13.3%	16.7	21.6	24.7	21.0	15.0%
Upper Bound										
(i) Multitask	10.3	14.1	25.7	16.7	46.0%	8.4	11.0	20.7	13.4	45.7%

A multitask learning upper bound was trained using all the three corpora jointly and simultaneously with results listed in the bottom row (i) of Table 1. This was expected to offer the best performance [41], since the LLL scenario assumes only minimal past data can be reused. Comparing to row (b) of Table 1, the multitask learning improved the ASR performance on WSJ and SWB (columns (i)(iii), row (i) of Table 1), while slightly degraded on LS (column (ii), row (i) of Table 1), showing that training E2E ASR with multiple datasets of very different distributions might not benefit to all domains, probably because the ASR model’s capacity was too small to generalize across many very different domains simultaneously. Still, multitask learning provided a good upper bound here otherwise.

3.3.2. Regularization-based Methods

We now inspect the results for regularization-based methods for CTC only without LM in Sec. (I) of Table 1. The methods EWC and SI (rows (c)(d)) both used relatively rigid constraints to limit each model parameter from drifting too far (Eq. (1)), except with different weights Ω_i^k . Compared to the trivial fine-tuning (row (b)), EWC offered the same or worse performance in all datasets compared to fine-tuning (rows (c) v.s. (b)). SI provided improvements for the first two datasets (columns (i)(ii) of row (d)) but failed to learn the last corpus (column (iii) of row (d)). The regularization methods here required model parameters to be close to the model for the first task (WSJ) but unable to help the model adapt to later tasks (LS and SWB). The scaling parameter λ in Eq. (1) may play some role here, but we found it challenging to tune and never successful.

In contrast, KD (row (e)) performed the best among the three regularization-based methods, reducing the catastrophic forgetting for both WSJ and LS (columns (i)(ii)) and offering good performance for the last task SWB (column (iii)). Also, giving a very good average or performance level considering both earlier and later tasks, significantly better than the fine-tuning (row (b)) baseline. The results were obviously due to the different concepts for regularization. Instead of constraining each parameter from drifting too far, KD tried to constrain the model output distributions close to the earlier models by KL

divergence. This method offered much more flexibility for the model parameters to drift freely while learning sequentially.

3.3.3. Data-based Methods

For data-based methods (GEM) (rows (f)(g)(h)), we allowed additional memory of 50MB, corresponding to roughly 30 minutes of audio data for 16bit 16kHz files and less than 1% of each corpus. Note that the regularization-based methods’ storage space was equivalent to the size of the CTC model, or 406MB here, so GEM required much less memory.

The results of GEM in row (f) of Table 1 outperformed all regularization-based methods, including the best one KD (row (e)), showing that storing a small dataset from the past and a good concept of learning from past data was useful. This is probably because learning from the past data offered more freedom for the model parameters to learn across the varying tasks. Compared to KD, the GEM-trained model reduced the catastrophic forgetting better on LS (column (ii), rows (f) v.s. (e)) while slightly worse on WSJ and SWB (columns (i)(iii)). This phenomenon is consistent with the previous finding that GEM was suitable for the earlier tasks but relatively weak for the most recent task, i.e., better backward transfer [11]. The higher freedom in shifting model parameters from stage to stage may end up with less precise parameters for later tasks or a kind of trade-off between earlier and later tasks.

Moreover, the proposed data selection PP and Len were similarly helpful and offered decent improvements over plain GEM (rows (g)(h) v.s. (f)). With GEM + Len, a relative reduction of 13.3% in averaged WER was obtained compared to the fine-tuning baseline (rows (h) v.s. (b)).

3.3.4. With Language Model Rescoring

A very similar trend as in Sec. (I) of Table 1 can be observed in Sec. (II) when an additional LM trained with all text data from the three corpora was applied for rescoring, demonstrating the achievable performance in our scenario. All WERR became higher (columns (x) v.s. (v)), showing that the multi-domain LM benefited ASR decoding on all three topic domains, and the performance gap among the three corpora was narrowed.

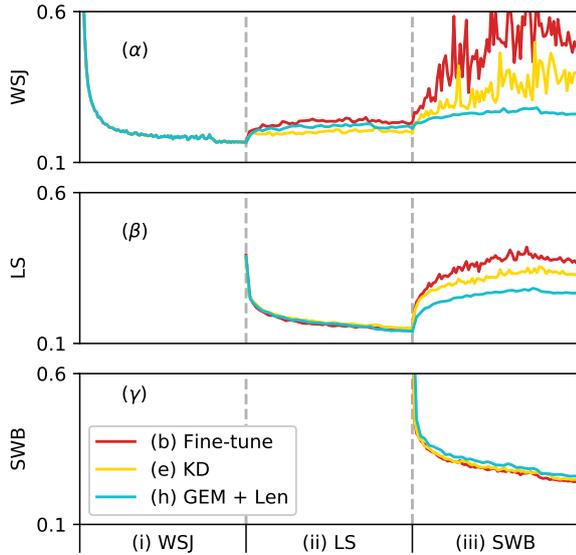


Figure 2: Learning curves for WERs of the CTC model under the training order of WSJ-LS-SWB, tested on (α) WSJ, (β) LS, and (γ) SWB.

3.4. Learning Curve

The learning curves (in WERs) of CTC models without RNN-LM, under the training order of WSJ-LS-SWB, are plotted in Fig. 2. We compared the best regularization- and data-based KD and GEM + Len approaches with the fine-tuning baseline in (rows (e)(h) v.s (b) of Table 1). The performance on WSJ, LS, and SWB of the CTC models during training are respectively shown in Sec. (α)(β)(γ) of Fig. 2. The horizontal scale is the training steps, with each stage normalized to the same width.

First, from Fig. 2(α), models were trained and tested on WSJ in the first stage, nothing happened, and the three curves merged into one. In the second and third stages of training on LS and SWB, however, all the three curves jumped up when switching the corpora and then tend to converge at higher levels, showing the phenomena of catastrophic forgetting. Inspecting the curves of the three methods, we found curves (e)(h) are significantly lower than curve (b), verifying KD and GEM + Len worked successfully here. In the third stage training with SWB, the results indicate that GEM + Len is much better than KD (curves (h) v.s. (e)). Moreover, curve (e) has a remarkably smaller amplitude of oscillation in the third stage, showing that exploiting a small amount of data from previous corpora stabilizes ASR training. Similar observations can be made in Fig. 2(β) tested on LS, where we start to record the learning process in the second stage of training on LS. For the last stage in Fig. 2(γ) trained and tested on SWB, GEM + Len performed slightly worse than the other two methods (curves (h) v.s. (b)(e)), consistent with the discussions on row (h) of Table 1, that is, GEM + Len is relatively weak for the most recent task.

3.5. Saved Data Size and Data Selection

Here we investigated the effect of different saved data sizes for GEM with randomly selected saved data and GEM + Len (rows (f)(h) of Table 1), all with LM applied. The results in Table 1 were for 30 minutes of saved data. We also tested with 5, 60, 120, and 220 minutes of saved data with the same setup, where

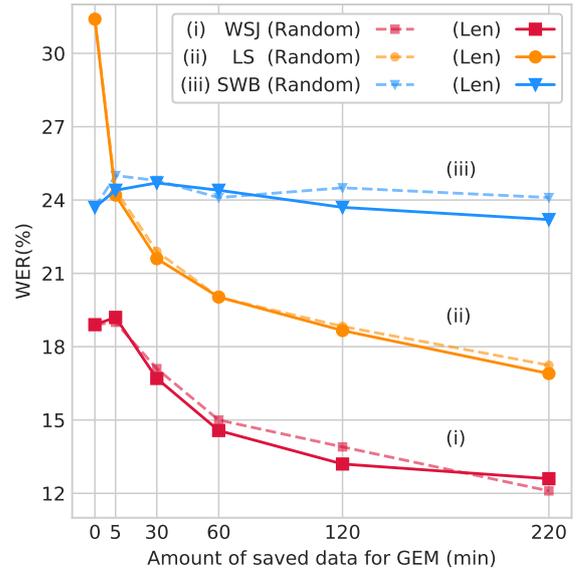


Figure 3: WERs with GEM under different saved data sizes, with random selection (dotted curves) or proposed Len (solid curves). Zero minutes of saved data is the fine-tuning baseline.

220 minutes of audio data is equivalent to the size of our CTC model. Results are shown in Fig. 3, where the leftmost points are the fine-tuning baseline (row (b), Sec. (II) of Table 1).

The general trend of improved performance with increased saved data size is clear, although not very apparent for SWB, showing more saved data lead to better backward transfer. The averaged WER over the three tasks for GEM + Len achieved was 17.6% WER (not shown in the figure) if 220 minutes of saved data were allowed, or a relative reduction of 28.7% than the fine-tuning baseline. Yet recognizing SWB is a difficult task, GEM + Len performed slightly better than fine-tuning when 220 minutes of past data were available. We found 5 minutes of saved data seemed insufficient for WSJ, however, both random selection and Len offered significant improvements for LS (orange curves). Moreover, in most cases, the proposed Len (solid curves) excelled the random selection (dotted curves), indicating an efficient data selection algorithm is attractive.

4. Conclusion

This is the first paper extensively exploring the feasibility and achievable performance of LLL for E2E ASR. We found data-based approaches were better, and proposed to properly select data from the past datasets by at least perplexity or utterance duration for mitigating catastrophic forgetting. The proposed methods can be easily applied to other E2E ASR frameworks like LAS or RNN-T. This is a small step towards the long term goal of having ASR technologies learning from the ever-changing real world incrementally.

5. Acknowledgements

We acknowledge the support of Salesforce Research Deep Learning Grant.

6. References

- [1] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, 1989, vol. 24.
- [2] Z. Chen and B. Liu, "Lifelong machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, 2018.
- [3] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, 2017.
- [4] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress & compress: A scalable framework for continual learning," in *ICML*, 2018.
- [5] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *ICML*, 2017.
- [6] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *ECCV*, 2018.
- [7] B. Ehret, C. Henning, M. R. Cervera, A. Meulemans, J. von Oswald, and B. F. Grewe, "Continual learning in recurrent neural networks with hypernetworks," *arXiv preprint arXiv:2006.12109*, 2020.
- [8] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.
- [9] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "Pathnet: Evolution channels gradient descent in super neural networks," *arXiv preprint arXiv:1701.08734*, 2017.
- [10] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *CVPR*, 2018.
- [11] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *NeurIPS*, 2017.
- [12] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, 2017.
- [13] F.-K. Sun, C.-H. Ho, and H.-Y. Lee, "LAMOL: LAnguage MOdeling for Lifelong Language Learning," in *ICLR*, 2019.
- [14] J. von Oswald, C. Henning, J. Sacramento, and B. F. Grewe, "Continual learning with hypernetworks," *ICLR*, 2019.
- [15] G. Saha and K. Roy, "Gradient projection memory for continual learning," *ICLR*, 2021.
- [16] J. von Oswald, C. Henning, J. Sacramento, and B. F. Grewe, "Continual learning with hypernetworks," *ICLR*, 2020.
- [17] D. Benavides-Prado, Y. S. Koh, and P. Riddle, "Towards knowledgeable supervised lifelong learning systems," *Journal of Artificial Intelligence Research*, vol. 68, 2020.
- [18] J. A. Mendez and E. Eaton, "Lifelong learning of compositional structures," *arXiv preprint arXiv:2007.07732*, 2020.
- [19] Y. Wen, D. Tran, and J. Ba, "Batchensemble: an alternative approach to efficient ensemble and lifelong learning," *ICLR*, 2020.
- [20] M. Rostami, D. Isele, and E. Eaton, "Using task descriptions in lifelong machine learning for improved performance and zero-shot transfer," *Journal of Artificial Intelligence Research*, vol. 67, 2020.
- [21] J. Xue, J. Han, T. Zheng, X. Gao, and J. Guo, "A multi-task learning framework for overcoming the catastrophic forgetting in automatic speech recognition," *arXiv preprint arXiv:1904.08039*, 2019.
- [22] S. Sadhu and H. Hermansky, "Continual learning in automatic speech recognition," *Interspeech*, 2020.
- [23] B. Houston and K. Kirchhoff, "Continual learning for multi-dialect acoustic models," *Interspeech*, 2020.
- [24] L. Fu, X. Li, and L. Zi, "Incremental learning for end-to-end automatic speech recognition," *arXiv preprint arXiv:2005.04288*, 2020.
- [25] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *ICML*, 2014.
- [26] A. Graves, "Sequence transduction with recurrent neural networks," in *ICML Workshop on Representation Learning*, 2012.
- [27] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016.
- [28] 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 *Interspeech*, 2020.
- [29] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation," in *ASRU*, 2017.
- [30] Z. Meng, Z. Chen, V. Mazalov, J. Li, and Y. Gong, "Unsupervised adaptation with domain separation networks for robust speech recognition," in *ASRU*, 2017.
- [31] G. I. Winata, S. Cahyawijaya, Z. Liu, Z. Lin, A. Madotto, P. Xu, and P. Fung, "Learning fast adaptation on cross-accented speech recognition," *arXiv preprint arXiv:2003.01901*, 2020.
- [32] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *HLT*, 1992.
- [33] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ASRU*, 2015.
- [34] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: telephone speech corpus for research and development," in *ICASSP*, 1992.
- [35] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
- [36] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *ICASSP*, 2013.
- [37] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *ASRU*, 2011.
- [38] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *ACL*, 2016.
- [39] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, 2019.
- [40] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Interspeech*, 2015.
- [41] T.-S. Nguyen, S. Stüker, and A. Waibel, "Toward cross-domain speech recognition with end-to-end models," in *Life-Long Learning for Spoken Language Systems Workshop - ASRU*, 2019.