Rainbow Keywords: Efficient Incremental Learning for Online Spoken Keyword Spotting

Yang Xiao, Nana Hou∗, Eng Siong Chng

School of Computer Science and Engineering, Nanyang Technological University, Singapore.

\{yxiao009,nana001\}@e.ntu.edu.sg, asechng@ntu.edu.sg

Abstract

Catastrophic forgetting is a thorny challenge when updating keyword spotting (KWS) models after deployment. This problem will be more challenging if KWS models are further required for edge devices due to their limited memory. To alleviate such an issue, we propose a novel diversity-aware incremental learning method named Rainbow Keywords (RK). Specifically, the proposed RK approach introduces a diversity-aware sampler to select a diverse set from historical and incoming keywords by calculating classification uncertainty. As a result, the RK approach can incrementally learn new tasks without forgetting prior knowledge. Besides, the RK approach also proposes data augmentation and knowledge distillation loss function for efficient memory management on the edge device. Experimental results show that the proposed RK approach achieves 4.2% absolute improvement in terms of Average Accuracy over the best baseline on Google Speech Command dataset with less required memory. The scripts are available on GitHub.

Index Terms: Incremental learning, Knowledge distillation, Online keyword spotting

1. Introduction

Spoken keyword spotting (KWS) [1] aims to identify the specific keywords in the audio input. It serves as a primary module in many real-world applications, such as Apple Siri and Google Home, which are widely utilized on the edge device [2, 3]. Current deep-learning-based keyword spotting systems are usually trained with limited keywords in the compact model for lower computation and smaller footprint [4, 5, 6]. Therefore, the performance of the KWS model trained by the source-domain data may degrade significantly when confronted with unseen keywords of the target-domain at run-time [7].

To alleviate such a problem, prior work [8, 9] utilize few-shot fine-tuning [10] to adapt KWS models with training data from the target-domain for new scenarios. However, performances on data from the source domain after adaptation could be poor, which is also known as the catastrophic forgetting problem [11]. Recent work [12] proposes a progressive continual learning [13, 14, 15] strategy for small-footprint keyword spotting to alleviate the catastrophic forgetting problem when adapting the model trained by source-domain data with the target-domain data. The limitations of such an approach are two-fold. First, the approach requires the task-ID as auxiliary information to learn the knowledge of different tasks, which is not always available in practice. Second, the storage volume occupied by the model will increase with the higher task numbers [16]. The storage volume will be unaffordable for light edge devices.

This paper proposes a novel diversity-aware incremental learning approach named Rainbow Keywords (RK) to address the issues mentioned above, requiring no task-ID information with fewer parameters. Specifically, the proposed RK approach introduces a diversity-aware sampler to select few but diverse examples from historical and incoming keywords by calculating classification uncertainty. As a result, the model will not forget the prior knowledge when learning new keywords even utilizing limited historical examples. Moreover, we utilize a mixed-labeled data augmentation to additionally improve the diversity of selected examples for higher performances. Besides, we propose a knowledge distillation loss function to guarantee that the prior knowledge could remain from the limited selected examples. We conduct our experiments on Google Speech Command dataset following the setup of prior work [17, 18]. Experimental results show that the proposed RK approach achieves 4.2% absolute improvement in terms of Average Accuracy over the best baseline with less required memory.

2. RK Architecture

With online keyword spotting systems, we assume that the model should identify all keywords in a series of tasks without catastrophic forgetting. For each task \(\tau_i\), we have input pairs \((x_i, y_i)\), where \(x_i\) denote audio utterances and \(y_i\) are keyword labels. We aim to minimize a cross-entropy loss \([19, 20]\) of all keywords \(N^t\) up to the current task \(\tau_i\) formulated as:

\[
L_{CE}(x, y) = \sum_{i=1}^{N^t} y_i \log \frac{\exp(o_i)}{\sum_{j=1}^{N^t} \exp(o_j)}
\]  

where \(o\) denotes the output logits of the model in the task \(\tau_i\).

2.1. Rainbow Keywords Network

We now introduce the proposed Rainbow Keywords (RK) approach, which consists of three parts: a diversity-aware sampler, a data augmentation strategy, and a knowledge distillation loss function. The detailed workflow of each module is as follows.

As illustrated in Figure 1, we first receive several incoming audio utterances of the task \(\tau_i\) as inputs \(D_{E}^{t−1}\), which are fed to the sampler together with historical examples \(\hat{D}_{E}^{t−1}\). Next, the diversity-aware sampler selects diverse examples \(D_{E}^{t}\) for task \(\tau_i\) by calculating classification uncertainty. Such selected examples \(D_{E}^{t}\) are then saved in examples storage and further augmented by the data augmentation to increase their diversity. Finally, we copy the old model as the teacher model and update the student model with augmented examples \(\hat{D}_{E}^{t}\) with the knowledge distillation loss function.
2.1.1. Diversity-Aware Sampler

The diversity-aware sampler aims to select diverse examples to manage memory efficiently. Such diverse examples are defined by the relative location of each example in feature space, which are estimated by the uncertainty of the sample through the inference by the classification model [21]. The required three steps are shown in the right red box of Figure 1:

Split by keywords: The first step gathers the historical examples \( D_{c,t-1} \) and incoming data \( D_{c,t} \), and groups them into subsets as \( D_c, c = 1 \ldots N^t \) by unique keywords, where \( N^t \) denotes the total numbers of unique keywords in \( D_{c,t-1} \cup D_{c,t} \) set.

Estimate uncertainty: The second step estimates the uncertainty of each sample \( x \) in \( D_c \) by Monte-Carlo (MC) method [22], which is defined in Equation 2.

\[
P(y = c \mid x) = \int p(y = c \mid \hat{x})p(\hat{x} \mid x)d\hat{x} \tag{2}
\]

where \( x, \hat{x}, y \) denote each audio utterance of keyword \( D_{N^t} \), the five perturbations of \( x \), and the label of \( x \). Therefore, the uncertainty of the audio utterance \( x \) is formulated as \( u(x) \):

\[
u(x) \approx 1 - \frac{1}{K} \sum_{k=1}^{K} P(y = c \mid \hat{x}_k) \tag{3}
\]

where \( K \) presents five perturbation strategies, including Clipping Distortion [23], TimeMask [23], Shift [24], PitchShift [24] and FrequencyMask [23]. The larger \( u(x) \) indicates that the less confidence of model to predict the perturbations.

Select by uncertainty: The third step selects \( L \) examples from \( D_c \) descending by uncertainty \( u(x) \) with the step size of \( \text{len}(D_c) \times N^t/L \). As a result, the most diversity examples are included in \( D_{c,t} \). Only these examples are available for training.

2.1.2. Data Augmentation

As the examples in \( D_{c,t} \) are few due to memory limitation, we apply the data augmentation to further increase the diversity [25] of \( D_{c,t} \). Specifically, we randomly mix two audio utterances to increase the amounts of training data without extra storage.

2.1.3. Knowledge Distillation Loss

Recent studies [26, 27, 28] show that Knowledge Distillation (KD) is effective for transferring knowledge between teacher-student models. Inspired by such theory, we consider the model of task \( \tau_{t-1} \) as the teacher model and the model of task \( \tau_t \) as the student model. We propose a knowledge distillation loss to preserve the prior knowledge from the teacher for the student model to avoid catastrophic forgetting, which is formulated as:

\[
\sigma(o^t_i(x); N^{t-1}) = \frac{\exp(o^i_t/T)}{\sum_{j=1}^{N^{t-1}} \exp(o^j_t/T)} \tag{4}
\]

\[
L_{KD}(o^t(x), o^{t-1}(x)) = \sum_{i=1}^{N^t} \sigma(o^i_t(x))\log(\sigma(o^{i-1}_t(x)))
\]

where \( o^{t-1}(x) \) and \( o^t(x) \) denote the output logits of the teacher model and student model, respectively. \( N^t \) is all keywords up to the task \( \tau_t \). \( \sigma \) is the knowledge distillation softmax function parameterized by the temperature \( T \). The temperature \( T \) is the...
experiential hyper-parameters of knowledge distillation set as 2.0. As a result, we aim to minimize the total loss of all keywords $N^t$ up to the current task $t$ formulated as:

$$L_{\text{total}}(x, y) = \lambda L_{CCE}(x, y) + (1 - \lambda)L_{KD}(o^t(x), o^{t-1}(x))$$

(5)

Where $L_{CCE}$ is the cross-entropy loss defined in Eq.1, and $L_{KD}$ is the knowledge distillation loss defined above. $\lambda$ is the experiential hyper-parameters defined as $\sqrt{1 - \frac{2^{t-1}}{N^t}}$.

### 3. Experiments and Results

#### 3.1. Dataset

We conduct experiments on the Google Speech Command dataset v1 (GSC) [29], which includes 64,727 one-second audio clips with 30 English keywords categories. We utilize 80% of data for training and 20% of data for testing. All of the audio clips in GSC are sampled at 16kHz in our experiment.

#### 3.2. Experimental Setup

##### 3.2.1. Network configuration

We employ the TC-ResNet-8 [30] as our testbed to evaluate the proposed rainbow keywords approach. It includes a 1-D convolution layer followed by three residual blocks, which consist of 1-D convolution, batch normalization and ReLU active function. Each layer has $16, 24, 32, 48$ channels.

We first pre-train the TC-ResNet-8 model on the GSC dataset with 32,000 audio clips, including 15 unique keywords. To evaluate the learning ability of the proposed RK approach, we split the rest data as 5 tasks. Each task includes 3 new unique keywords, which is unseen in previous tasks. To simulate the condition of edge devices, we set the max amount of examples $L$ due to the limited memory in edge devices [17, 18].

During the training stage, we utilize the Mel-frequency cepstrum coefficients (MFCC = 40) as inputs.

##### 3.2.2. Reference baselines

We built eight baselines for comparisons.

- **Fine-tune training**: adapts the TC-ResNet-8 model for each new task without class incremental learning (CIL) strategies. We consider it as the lower-bound baseline.

- **NR [31]**: is a CIL approach which randomly selects training samples from previous tasks for future training.

- **iCaRL [28]**: is a CIL approach which selects the samples close to the mean of its own class. Then iCaRL utilizes examples for future training.

- **EWC [32]**: is a CIL approach which incorporates a quadratic penalty to regularize parameters of model that were important to past tasks. The importance of parameters is approximated by the Fisher Information Matrix.

- **RWalk [33]**: is a CIL approach which improves the EWC. Both Fisher Information Matrix approximation [32] and online path integral [34] are fused to calculate the importance for each parameter. RWalk also selects historical examples and utilizes them for future training.

- **BiC [27]**: is a recent CIL approach with more attentions. BiC introduces an additional layer to correct task bias of the network. BiC also uses the same sampling method as iCaRL to select historical examples.

- **PCL-KWS [12]**: is a CL approach for Spoken Keyword Spotting. Specifically, the PCL-KWS includes several task-specific sub-networks to memorize the knowledge of the previous keywords. Then, a keyword-aware network scaling mechanism is introduced to reduce the network parameters. The PCL-KWS requires the task-ID to select corresponding sub-networks.

- **Joint training**: trains the TC-ResNet-8 model with the whole dataset, regardless of any constrains. We consider it as the upper-bound baseline.

#### 3.3.3. Effect of various memory size on RK approach

We further analyse and summarize the performances of the proposed knowledge distillation loss function achieves 54.5% relative improvements in terms of BWT.

- **PCL-KWS [12]**: is a CL approach for Spoken Keyword Spotting. Specifically, the PCL-KWS includes several task-specific sub-networks to memorize the knowledge of the previous keywords. Then, a keyword-aware network scaling mechanism is introduced to reduce the network parameters. The PCL-KWS requires the task-ID to select corresponding sub-networks.

- **Joint training**: trains the TC-ResNet-8 model with the whole dataset, regardless of any constrains. We consider it as the upper-bound baseline.

- **PCL-KWS [12]**: is a CL approach for Spoken Keyword Spotting. Specifically, the PCL-KWS includes several task-specific sub-networks to memorize the knowledge of the previous keywords. Then, a keyword-aware network scaling mechanism is introduced to reduce the network parameters. The PCL-KWS requires the task-ID to select corresponding sub-networks.

#### 3.3. Results

##### 3.3.1. Effect of the knowledge distillation loss

We first analyse and summarize the performances with knowledge distillation loss. As shown in Table 1, we observe that the proposed knowledge distillation loss function achieves 54.5% relative improvements in terms of BWT.

##### 3.3.2. Effect of various task numbers on RK approach

We first analyse and summarize the performances of the proposed RK approach with increasing task numbers as shown in Figure 2. The x-axis represents the task numbers ($= 20, 10, 5, 4$) and the y-axis is the evaluation metric of ACC. For example, when the task number is set to 20, we pre-train the TC-ResNet-8 with 21,000 audio clips including 10 unique keywords and the rest data is split into 20 tasks including 1 unique unseen keyword. The corresponding ACC is the accuracy of the testing set after each task is finished training. The memory size $L$ is 1500 for the proposed RK approach here. We observe that the proposed RK approach achieves the best ACC performances with increasing task numbers. Even though the difficulty of preserving prior knowledge is increased with the increasing task numbers, the proposed RK approach still can obtain over 85.0% ACC, which is much better than other baselines.

##### 3.3.3. Effect of various memory size on RK approach

We then report the effect of various memory size ($L=300$ or $500$) on RK approach, as shown in Figure 3. We constrain all methods under the same memory size as that of the RK approach. We observe that even all approaches perform much
### 3.3.4. Effect of the data augmentation

We also summarize the performances of the effects with two data augmentation methods on the proposed RK approach, as shown in Table 2. We observe that all data augmentation methods improve the performances in terms of ACC. The best performances are achieved by the ‘Mixup’ data augmentation. We adopt the ‘Mixup’ data augmentation hereafter.

![Figure 2: The ACC (%) in a comparative study of increasing task numbers on the proposed RK approach and other competitive baselines. Figures from (a) to (d) represent the experiment with task numbers (= 20, 10, 5, 4).](image)

![Figure 3: The ACC (%) in a comparative study of various memory size on the proposed RK approach and other competitive baselines.](image)

Table 2: Average Accuracy (ACC) and Backward Transfer (BWT) in a comparative study of the proposed data augmentation. “NoAugment” means no data augmentation is applied in the experiment. L denotes the memory size for RK.

| Methods (L=500) | ACC(↑) | BWT(↑) |
|-----------------|--------|--------|
| NoAugment       | 0.779  | -0.033 |
| SpecAugment [23]| 0.783  | -0.006 |
| Mixup [36]      | 0.828  | -0.036 |

better with increasing memory size (i.e., more available training data), the proposed RK approach still outperforms other methods. Furthermore, even with limited memory size (L=300), the proposed RK approach can obtain over 65.0% ACC performances, much better than other methods.

### 3.3.5. Benchmark against other competitive methods

Table 3 summarizes the comparison between the proposed RK and other competitive methods in terms of Average Accuracy (ACC), Backward Transfer (BWT), Parameters and Memory. The task number is set to 5. We observe that the proposed RK achieves the best performance with memory size L of 3000. Comparing with the best baselines: RWalk, the proposed RK-3000 achieves 4.2% absolute improvements in terms of Average Accuracy with fewer memory, which is closer to the upper-bound performances. Furthermore, even with only 16.2M training data, our approach RK-500 has comparable performance to other baseline methods, which is effective on edge devices.

![Figure 2: The ACC (%) in a comparative study of increasing task numbers on the proposed RK approach and other competitive baselines. Figures from (a) to (d) represent the experiment with task numbers (= 20, 10, 5, 4).](image)

![Figure 3: The ACC (%) in a comparative study of various memory size on the proposed RK approach and other competitive baselines.](image)

Table 3: Accuracy and efficiency metrics in a comparative study of recent state-of-the-art training strategies. We adopt the TC-ResNet-8 model as testbed for all training strategies for fair comparison. Memory size L is set to [500, 1500, 3000] in following experiments for RK.

| Methods | ACC(↑) | BWT(↑) | Parameters | Memory |
|---------|--------|--------|------------|--------|
| Fine-tune | 0.262  | -0.372 | 64.48K     | 16.2M  |
| EWC     | 0.835  | -0.064 | 129.96K    | 16.2M  |
| PCL-KWS | 0.836  | -0.041 | 406.9K     | 16.2M  |
| NR      | 0.560  | -0.163 | 64.48K     | 178.6M |
| iCaRL   | 0.846  | -0.057 | 75.29K     | 178.6M |
| BiC     | 0.793  | -0.085 | 64.48K     | 178.6M |
| RWalk   | 0.871  | -0.045 | 129.96K    | 178.6M |
| RK-500  | 0.828  | -0.036 | 129.96K    | 16.2M  |
| RK-1500 | 0.887  | -0.012 | 129.96K    | 48.6M  |
| RK-3000 | 0.913  | -0.012 | 129.96K    | 97.2M  |
| Joint   | 0.940  | -       | 64.48K     | 1624.4M|

In this paper, we propose a novel diversity-aware class incremental learning method named Rainbow Keywords (RK) approach to avoid catastrophic forgetting with less memory. Experimental results show that the proposed RK approach achieves 4.2% absolute improvement in terms of average accuracy over the best baseline. Ablation study also indicates that the proposed data augmentation and knowledge distillation loss are quite effective on edge devices. For future work, we plan to apply and adapt our approach to multilingual keyword search systems [37, 38, 39].

### 4. Conclusions

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-100E-2018-006).

The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore (https://www.nscc.sg).
5. References

[1] I. López-Espejo, Z.-H. Tan, J. Hansen, and J. Jensen, “Deep spoken keyword spotting: An overview,” 2021.

[2] Y. Zhang, N. Suda, L. Lai, and V. Chandra, “Hello edge: Keyword spotting on microcontrollers,” 2018.

[3] H. Zhang, Y. Li, Y. Huang, Y. Wen, J. Yin, and K. Guan, “Mimodel: An automatic cloud platform for efficient mlas,” in ACM Multimedia. ACM, 2020, pp. 4453–4456.

[4] G. Chen, C. Parada, and G. Heigold, “Small-footprint keyword spotting using deep neural networks,” in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014, pp. 4087–4091.

[5] A. Berg, M. O’Connor, and M. T. Cruz, “Keyword transformer: A self-attention model for keyword spotting,” arXiv preprint arXiv:2104.00769, 2021.

[6] B. Kim, S. Chang, J. Lee, and D. Sung, “Broadcasted residual learning for efficient keyword spotting,” arXiv preprint arXiv:2106.04140, 2021.

[7] N. Hou, C. Xu, V. T. Pham, J. T. Zhou, E. S. Chng, and H. Li, “Speaker and phoneme-aware speech bandwidth extension with residual dual-path network,” in INTERSPEECH. ISCA, 2020, pp. 4964–4968.

[8] A. Awasthi, K. Kilgour, and H. Rom, “Teaching keyword spots to spot new keywords with limited examples,” arXiv preprint arXiv:2106.02443, 2021.

[9] M. Mazumder, C. Banbury, J. Meyer, P. Warden, and V. J. Reddi, “Few-shot keyword spotting in any language,” Interspeech 2021, Aug 2021. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2021-1966

[10] Y. Li, H. Zhang, S. Jiang, F. Yang, Y. Wen, and Y. Luo, “Modelups: An interactive and collaborative platform for editing pre-trained models at scale,” CoRR, vol. abs/2105.08275, 2021.

[11] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in CONNECTIONIST NETWORKS: The Sequential Learning Problem, in Proceedings of the national academy of sciences, vol. 114, no. 13, pp. 3521–3526, 2017.

[12] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, “Riemannian walk for incremental learning: Understanding forgetting and intransigence,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 532–547.

[13] F. Zenke, B. Poole, and S. Ganguli, “Continual learning through synaptic intelligence,” in International Conference on Machine Learning. PMLR, 2017, pp. 3987–3995.

[14] D. Lopez-Paz and M. Ranzato, “Gradient episodic memory for continual learning,” Advances in neural information processing systems, vol. 30, 2017.

[15] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” arXiv preprint arXiv:1710.09412, 2017.

[16] N. F. Chen, S. Vidas, C.-H. Lee, E. S. Chng, B. Ma et al., “Exemplar-inspired strategies for low-resource spoken keyword search in swahili,” in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016, pp. 6040–6044.

[17] N. F. Chen, S. Vidas, B. P. Lim, H. G. Ngo, H. Xu, B. Ma, H. Li et al., “Strategies for vietnamese keyword search,” in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014, pp. 4121–4125.

[18] N. F. Chen, C. Ni, I.-F. Chen, S. Vidas, H. Xu, X. Xiao, T. S. Lau, S. J. Leow, B. P. Lim, C.-C. Leung et al., “Low-resource keyword search strategies for tamil,” in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015, pp. 5366–5370.