CS753
Course Project

Efficient Continual Learning for Keyword Spotting

Anugole Sai Gaurav, 170070008
Anwesh Mohanty, 170070009
Jian Vora, 170100026

Problem Statements

1. Joint Keyword Spotting and Speaker Identification: The problem can be considered as a binary classification task where at test time, if an utterance matches with an already enrolled keyword and is said by the same speaker then output 1, else output a 0.
2. Continual Batch Keyword Spotting: We have multiple datasets, each having a different set of keywords. At each time step we only have access to one of these datasets. Our model (before the softmax) should be able to adapt to this new split while still performing well on the previous data.

Joint KWS and Speaker Identification

Joint KWS and Speaker Identification

1. Started off with the Interspeech Challenge
2. **Dataset Description:**

Total Number of Speakers: 100

Utterances per speaker of the keyword: 10

Utterances per speaker of any general word (may include keyword): 30

Each speaker had a different keyword and a different language as well, and we had a binary classification task:

a. Output 1 if the keyword is correct and uttered by the same speaker
b. Ouptut 0 in all other cases

Methodology

We started off with the simple baseline of using a cascaded model of independently trained models for keyword spotting and speaker identification

Keyword Spotting

Due to the dearth of labelled data, we used transfer learning using the TC-ResNet backbone trained on Speech Commands and training the final softmax layer only

Speaker Verification

We use SincNet which is basically a CNN model with early layers replaced by Sinc filters. We explore both transfer learning and e2e training with two types of loss functions as described subsequently.

Results

Keyword Spotting

Train Accuracy: **98.12**, Validation Accuracy: **85.64**

Speaker Identification ===> Transfer Learning Results

Train Accuracy: 98.12, Validation Accuracy: 85.64

Speaker Identification ===> End-to-End Training Results

Train Accuracy SincNet: 86.53, AM-SincNet: **99.74**

Valid Accuracy SincNet: 72.74, AM-SincNet: **86.17**

Final Resultant System

The algorithm is as follows: if(keyword not identified): return 0

elif(keyword and speaker identified and match): return 1

else: return 0

The train accuracy (over 100 classes) is **88.56**

The valid accuracy (over 100 classes) is **59.34**

Steps Planned Earlier

Try to formulate the problem as that of multi-task learning (2 tasks at hand) and perform a joint training

Realised we had misinterpreted the problem statement

At test time, we had to enroll a new speaker in the system with a few training utterances which could not be handled by the work done by us earlier

Hence, a change in the problem statement ...

Continual Batch KWS

Model Architecture

- 5 conv layer with residual connections: batch norm, dropout (0.1)
- 2 Fully connected layers
- Final softmax layer

Dataset

- We have used the Speech Commands Dataset for this project
- It contains 105,829 one-second (or less) utterances of 35 different words
- The training and testing splits have been made according to the splits in the implementation of the dataset in torchaudio
- In the first set of experiments we break the 35 words into 2 different sets:
  - 2 Domains A and B: A has 15 classes and B has 20 classes
  - 3 Domains A, B and C: A has 12 classes, B has 12 classes and C has 11 classes
- In the next set of experiments we consider 3 domains with 10 classes each and the samples of 5 remaining classes are used as unknown labels

Domain Adaptation (DA) Methods

1. Elastic Weight Consolidation (EWC): Weighted L2 penalty between parameters of the two models with weights coming from the diagonal of the Fisher Information matrix.

$$\tilde{L}(\theta) = L_{new}(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta^*_{old,i})^2$$

2. Learning Without Forgetting (LWF): Additional L2 penalty on the features generated by the old and new classifier on current domain.

$$\tilde{L}(\theta) = L(\theta_{new}) + \lambda \sum_{X_i} (\theta_{new}(X_i) - \theta_{old}(X_i))^2$$

3. Continual Replay Adaptation (CRA): Additional replay loss (CE loss) applied on the predictions made by current model and the old model on a subset of samples from old domain.

$$\tilde{L}(\theta) = L(\theta_{new}) + \lambda L_{replay}(C(FE(X_p)), Y_p)$$

Experiment 1

In this experiment, we maintain a common feature extractor to predict on all the test datasets. As mentioned previously, we divide the Speech Commands dataset into 2 or 3 classes and continually train on the classes sequentially. The assumption here is that we know from which dataset the sample is coming from during the test phase.

- For example, first we train our model on dataset A containing 15 classes.
- Then we initialize the model for dataset B by using the feature extractor of dataset A (remove the last softmax layer).
- Then we train the model on B using our DA methods, and during test phase use this model for both dataset A and B.
- Repeat the process if more datasets exist.

Results of Experiment-1

With 2 datasets: Dataset A (15 classes) followed by Dataset B (20 classes).

Before adaptation, accuracy on A is 86.78%.

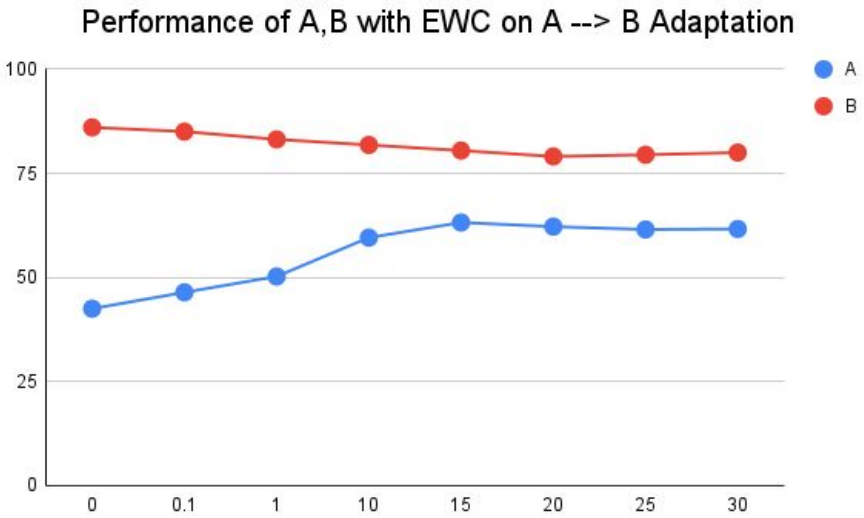
Method	After Adaptation	
	A	B
Baseline	42.51	86.04
EWC (λ = 15)	63.2	80.48
LWF (λ = 0.0005)	44.5	86
CRA (λ = 1)	53.17	86.52

With 3 datasets: Dataset A (12 classes), B (12 classes) and C (11 classes). The order followed is A->B->C.

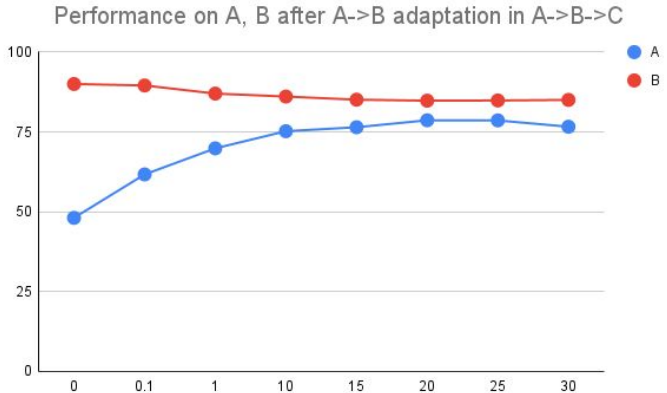
Before adaptation, accuracy on A is 89.26%.

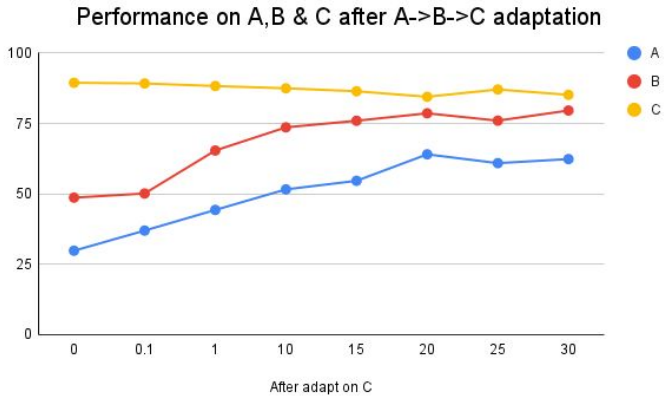
Method	A → B		A → B → C		
	A	B	A	B	C
Baseline	48.1	90.1	29.82	48.67	89.45
EWC (λ = 20)	78.65	84.79	64.05	78.62	84.48
LWF (λ = 0.0005)	62.56	91.68	37.42	62.77	91.14
CRA (λ = 1)	61.34	82.01	49.29	62.35	88.36

Parameter Tuning for EWC



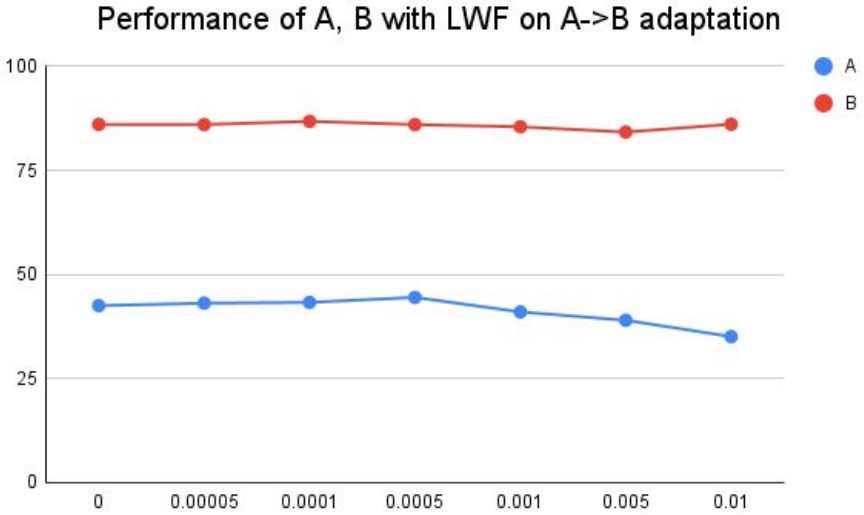
For 2 datasets, after λ = 15 the performance on A saturates. Hence we choose λ = 15 for our experiments.



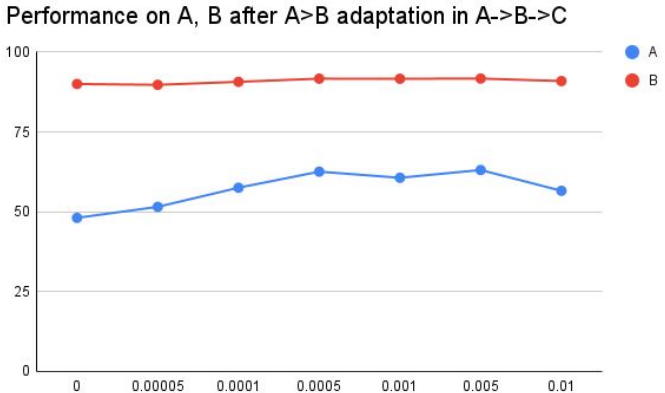


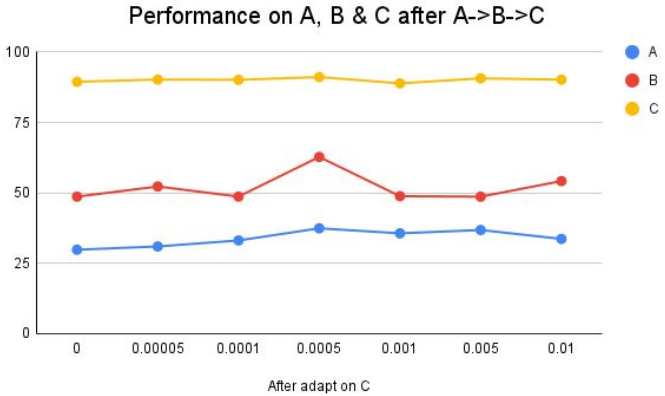
For 3 datasets, after λ = 20 the performance on A,B and C approximately saturates. Hence we choose λ = 20 for our experiments.

Parameter Tuning for LWF



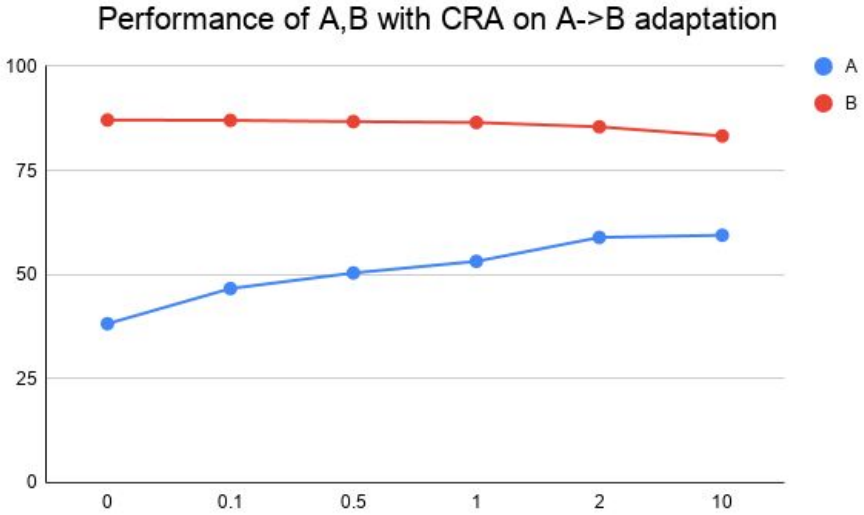
For 2 datasets, there is a small increase in accuracy of A till λ = 0.0005, after that it deteriorates. Hence we choose λ = 0.0005 for our experiments.



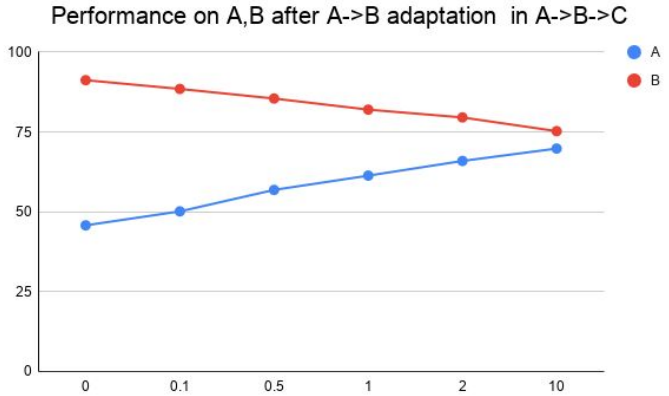


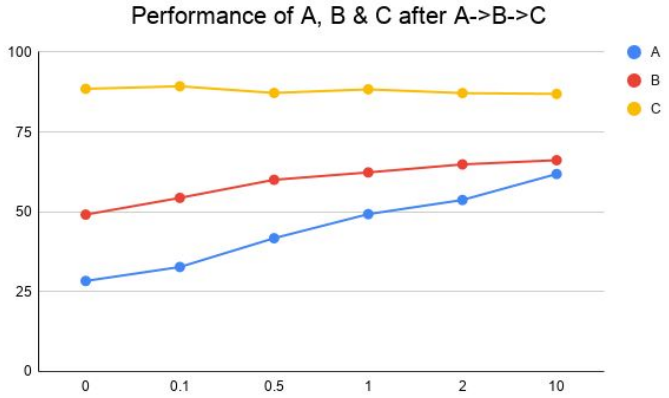
For 3 datasets, after λ = 0.0005 the performance on A,B and C approximately saturates/falls. Hence we choose λ = 0.0005 for our experiments.

Parameter Tuning for CRA



There is a tradeoff between performances of A and B with λ. We have chosen λ = 1 as after that we noticed slightly unstable performances.





For 3 datasets, after λ = 1 the performance on A,B increases but C decreases.Hence we choose λ = 1 as a middle point for our experiments.

Observations

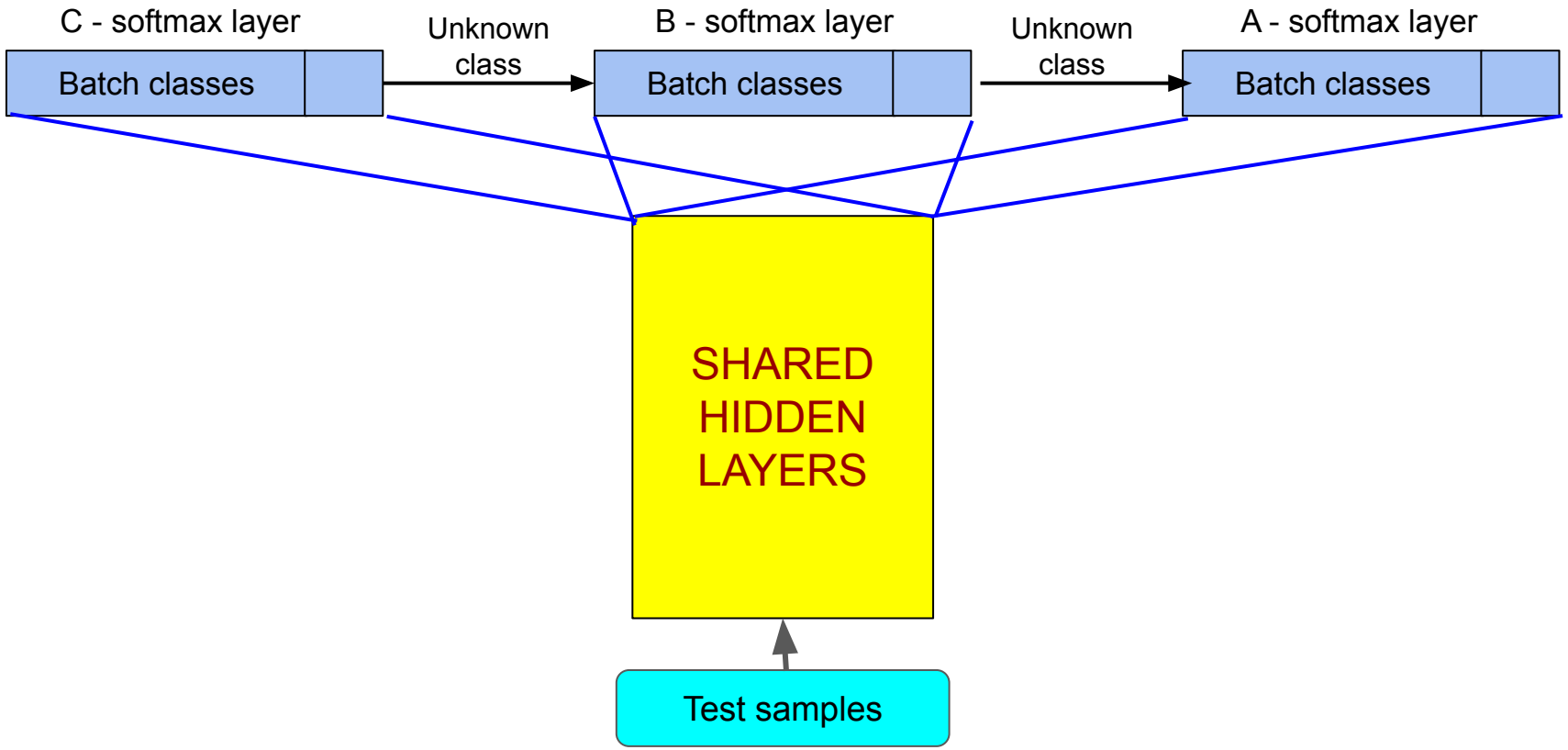
- Applying EWC during adaptation leads to significantly better results compared to the baseline. The current model is able to perform well on the current dataset as well as the older datasets.
- LWF doesn’t work very well in this setup as can be seen from the results and parameter tuning graphs. There is only a marginal increase in performance at most times.
- CRA works quite well but the results are still inferior compared to EWC. The performance on older domains increases with λ but the model becomes more unstable as λ is kept on increasing.
- We use the optimal parameters (λ) obtained in this part in the next experiment.

Experiment 2

We do away with our previous assumption that we know to which dataset a sample belongs to at test time. So the model at any time instant can receive samples from classes that are not part of the current classes that it is training on. For this setup, we increase the number of classes at any step by 1, to account for the “unknown” class. A sample example looks like:

- Suppose dataset A has 12 classes. We train a model with 13 classes and use samples from other classes (not in the 12) and give them “unknown” label.
- We use our previously defined DA methods to train the model on dataset B.
- During test time now, a sample first goes through the feature extractor and softmax layer of B and checks if it is unknown or not. If it is unknown, then we check it through the softmax layer of A to see if belongs to dataset A.

Model Architecture for combined testing



Combined Testing results

	EWC	LWF	CRA
A → B			
AB	69.67	61.24	71.34
B	82.29	86.11	84.37
A	58.06	47.51	57.18
A → B → C			
ABC	53.43	41.28	56.38
C	79.99	65.10	68.18
B	40.82	38.17	45.86
A	37.43	34.36	36.27

Observations

- With the combined testing approach used, the accuracies on test dataset becomes lower as more batches are adapted.

- Using the same approach, the accuracies on individual batch samples are also lower compared to Expt 1 due to increasing possibility of misclassification, with recently adapted batch having the highest accuracy.

- On average, the accuracy metrics for EWC and CRA loss seems to perform better than LWF for same adaptation scenarios and test data.

Conclusions

- From Expt1 and Expt2 it is clear that the initial assumption is a very crucial one; if we know to which domain a test sample belongs to then the model performance is much better
- EWC in general tends to perform better than CRA and LWF, it performs best under conditions of expt1
- Though the DA methods reduce the amount of forgetting by a significant margin, it still happens and might become significant with more domains
- **Possible future work:** Instead of making an unknown class, it might be better to have a discriminator while combined testing to find out to which domain a sample belongs to.

Contributions

Gaurav:

- KWS task on TC-Resnet
- Combined testing for EWC

Anwesh:

- Research on domain adaptive techniques, state-of-the-art KWS networks
- Implementation of EWC, LWF and CRA methods
- Combined testing for LWF

Jian:

- Speaker Verification using SincNet; Joint training for verification and KWS
- Combined testing for CRA

THANK YOU