CS771 Mini project 2

Devank Saran Prajapati
220339

Mudit Jain
220674

Pranjul Shikhar Verma
220797

Rashi Sharma
220872

Riyanshi
220903

Yuvan Gorani
221233

1 Introduction

Our project focuses on continual learning to address the challenge of training models iteratively on a sequence of datasets with shifting input distributions. For problem 1, starting with a single labeled dataset (D_1) and leveraging pseudo-labeling techniques, we are required to build models that generalize effectively across related but distinct datasets (D_2-D_{20}). D_1 to D_{10} share the same input distribution ($p(x)$) while D_{11} to D_{20} have distinct but related input distributions. The key goals are to maintain high accuracy on current and past datasets while preventing catastrophic forgetting, ensuring robust model updates without performance degradation.

For problem 2 of this project, we read the research paper Lifelong Domain Adaptation via Consolidated Internal Distribution [1] and presented a video explaining the idea proposed in the paper and a comparative study with existing algorithms.

2 Problem 1

2.1 Task 1

The solution employs a **prototypical learning** method with **vision transformers (ViT)** as the feature extractor. The approach is outlined as follows:

i) Feature Representation: Each image is processed through a Vision Transformer (ViT) to extract a 768-dimensional feature vector. These vectors are used to create class-wise prototypes.

ii) Prototype creation: For each class, a prototype is computed as the mean feature vector of all images belonging to that class:

$$Prototype_c = \frac{1}{N_c} \sum_{i \in C_c} Feature_i$$

where N_c is the number of images in class c, and C_c is the set of all images in class c.

iii) Classification: At test time, a test image's feature vector is compared to all prototypes using Euclidean distance. The image is assigned to the class of the closest prototype:

$$Class = \arg\min_c \|Feature_{test} - Prototype_c\|_2$$

2.1.1 Approach 1: Sequential prototype updating

1. Compute initial prototypes f_1 using the labeled dataset D_1.
2. Use f_1 to predict pseudo-labels for D_2, then compute updated prototypes f_2.
3. Repeat the process for datasets D_3 to D_{10}, updating the prototypes iteratively: $f_1 \rightarrow f_2 \rightarrow \cdots \rightarrow f_{10}$.
4. Evaluate each model f_i on its corresponding held-out dataset H_i and all prior held-out datasets H_j ($j < i$).

2.2 Approach 2: Weighted prototype averaging

1. For each dataset D_{i+1}, compute the current prototype ($prototype_i$) using pseudo-labels.

2. Combine it with the previous prototype ($prototype_{i-1}$) to form the final prototype:

$$Final\_Prototype = x \cdot prototype_i + (1 - x) \cdot prototype_{i-1}$$

3. Test various values of x ($x = 0, 0.1, 0.2, \ldots, 0.9$) and select the one achieving the highest accuracy. We finally used the value of x = 0.9 for best results.
4. Use the final prototype for pseudo-labeling the next dataset and repeat the process.

2.2.1 Evaluation

The performance of each model f_i is evaluated on its held-out dataset $\hat{D}_i$ and all previous held-out datasets ($\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_{i-1}$). Results are presented in the form of an accuracy matrix:

Model	$\hat{D}_1$	$\hat{D}_2$	$\hat{D}_3$	$\hat{D}_4$	$\hat{D}_5$	$\hat{D}_6$	$\hat{D}_7$	$\hat{D}_8$	$\hat{D}_9$	$\hat{D}_{10}$
f_1	94.16	–	–	–	–	–	–	–	–	–
f_2	92.4	92.04	–	–	–	–	–	–	–	–
f_3	90.6	90.2	91	–	–	–	–	–	–	–
f_4	90	90.2	90.56	91.04	–	–	–	–	–	–
f_5	89.28	89.48	89.92	89.96	89.4	–	–	–	–	–
f_6	89.24	89.52	89.96	89.96	89.36	89.8	–	–	–	–
f_7	89.12	89.32	89.72	90	89.4	89.68	89.68	–	–	–
f_8	89.08	89.08	89.4	90.2	89.2	89.68	89.68	89.08	–	–
f_9	88.96	88.96	89.68	89.4	88.72	89.6	89.68	89	89.28	–
f_{10}	88.36	88.52	88.96	89.12	88.48	89.2	89.28	89	88.76	89.52

Table 1: Accuracy matrix showing performance of models on held-out datasets via sequential updating approach.

Model	$\hat{D}_1$	$\hat{D}_2$	$\hat{D}_3$	$\hat{D}_4$	$\hat{D}_5$	$\hat{D}_6$	$\hat{D}_7$	$\hat{D}_8$	$\hat{D}_9$	$\hat{D}_{10}$
f_1	94.16	–	–	–	–	–	–	–	–	–
f_2	94.04	94.24	–	–	–	–	–	–	–	–
f_3	93.8	94.12	94.08	–	–	–	–	–	–	–
f_4	93.76	94.04	93.92	94.36	–	–	–	–	–	–
f_5	93.64	93.84	93.8	94.36	93.84	–	–	–	–	–
f_6	93.36	93.6	93.6	94.16	93.64	94.12	–	–	–	–
f_7	93.16	93.28	93.52	94	93.48	94	93.72	–	–	–
f_8	92.92	93.08	93.36	93.84	93.36	93.92	93.64	93.36	–	–
f_9	92.8	92.76	93.24	93.68	93.24	93.68	93.6	93.12	93.28	–
f_{10}	92.64	92.6	93.2	93.6	93.08	93.52	93.6	92.92	93	93.68

Table 2: Accuracy matrix showing performance of models on held-out datasets via weighted prototype approach.

2.2.2 Insights

We tried two different approaches, sequential updating approach and weighted prototype approach. The weighted prototype averaging approach outperforms the sequential updating approach by maintaining consistent accuracy across datasets. It mitigates the impact of pseudo-label errors and ensures better retention of information from earlier prototypes. Future improvements could include dynamic selection of the weighting factor x or regularization techniques to refine prototype updates.

2.3 Task 2

Due to the difference in the distributions for datasets 11 to 20, unlike datasets 1 to 10, datasets 11 to 20 offer high level of shock to the model while performing predictions on those sets.

2.3.1 Approach

The approach of task 2 is similar to that of task 1, here also we used weighted prototype approach but with dynamic selection of weights for different models.

2.4 Saved prototypes

Since prototype learning and feature extraction are time-consuming processes, we have saved the learned prototypes used in task 1 and the features used in task 2 in the respective folders.

2.4.1 Evaluation

Data Model	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
11	92.44	92.32	92.76	92.92	92.72	93.24	93.24	92.52	92.72	93.24	74.72	-	-	-	-	-	-	-	-	-
12	91.44	90.68	91.68	92.00	90.96	92.00	91.76	91.16	91.56	91.92	73.32	66.16	-	-	-	-	-	-	-	
13	91.48	90.76	91.88	92.08	90.96	92.00	91.76	91.36	91.56	91.84	73.08	65.84	80.04	-	-	-	-	-	-	
14	91.28	90.64	91.88	92.08	91.08	92.00	91.72	91.28	91.68	91.80	72.84	65.20	79.72	85.44	-	-	-	-	-	-
15	91.24	90.48	91.80	91.92	90.96	91.96	91.68	91.36	91.56	91.68	72.36	65.00	79.52	85.32	88.92	-	-	-	-	-
16	90.52	89.92	91.12	91.12	91.08	91.40	91.32	90.24	90.96	91.16	71.48	62.64	78.44	84.68	88.52	78.80	-	-	-	-
17	90.76	90.20	91.08	91.44	91.00	91.56	91.32	90.28	91.08	91.28	71.44	61.88	78.44	85.00	88.68	78.16	71.96	-	-	-
18	90.28	89.96	91.24	91.20	90.24	91.40	91.16	90.44	90.92	91.24	71.32	62.12	78.36	84.48	88.44	77.64	71.56	76.12	-	-
19	89.16	88.64	89.68	89.36	88.84	89.28	89.72	89.20	89.64	89.32	68.36	59.00	76.72	82.64	86.96	75.00	68.12	72.92	71.36	-
20	89.60	89.04	90.00	89.92	89.36	89.76	90.28	89.20	90.04	89.92	68.88	58.88	76.36	82.76	87.08	75.32	68.84	72.72	70.00	82.00

Table 3: Accuracy matrix showing performance for models 11-20 over datasets from 1 to 20.

3 Problem 2

Youtube link : https://youtu.be/aTopcRbn7k4

References

[1] Mohammad Rostami. Lifelong domain adaptation via consolidated internal distribution. *Advances in Neural Information Processing Systems (NeurIPS 2021)*, 2021.