Indian Institute of Technology Kanpur

INTRODUCTION TO MACHINE LEARNING (CS771)

MINI-PROJECT 2 REPORT

Students Name	Student ID
Sneha Barman	221068
Shrasti Sahu	221025
Tushar Sahu	221146
Atul Kumar Bhongade	220252

Instructor:

Dr. Piyush Rai

Submission Date : 26/11/2024

1 Introduction

The focus of this work is to design a robust Learning with Prototypes (LwP) classifier capable of addressing challenges related to pseudo-label generation and domain adaptation. The iterative framework trains models on a sequence of datasets that share similar or slightly varying distributions while mitigating catastrophic forgetting.

This project addresses the challenges of incremental learning over multiple subsets of CIFAR-10. The task involves using the Learning with Prototypes (LwP) approach to iteratively train models $f_1, f_2, \ldots, f_{20}$ on datasets D_1 to D_{20}. The goal is to maximize classification accuracy on the held-out labeled sets for each dataset while minimizing performance degradation on earlier datasets (*catastrophic forgetting*) and maintaining robust generalization across varying distributions (*domain shift*). Using a pretrained ResNet50 for feature extraction and leveraging concepts from the "*Deja Vu : Continual Model Generalization for Unseen Domains*" paper, we systematically update models to maintain performance consistency across prior datasets.

2 Problem Statement

The problem involves a sequence of 20 datasets, $\{D_1, D_2, \ldots, D_{20}\}$, where:

- D_1 is fully labeled and serves as the initial training set.
- D_2 to D_{20} are unlabeled, with D_2 to D_{10} sharing the same distribution as D_1, and D_{11} to D_{20} belonging to different but related distributions.

The primary task is to:

1. Generate reliable pseudo-labels for the unlabeled datasets $\{D_2, \ldots, D_{20}\}$.
2. Iteratively train models $\{f_1, f_2, \ldots, f_{10}\}$ on the datasets, ensuring minimal performance degradation on previously trained datasets.
3. Checking the performance of the models on previously seen datasets via held-out labeled sets.

Several constraints govern the solution:

- Neural networks or advanced models are not permitted for training; the solution must rely solely on LwP methodologies.
- Only pretrained neural network may be used for feature extraction.

Challenges

Pseudo-Labeling Unlabeled Data

The lack of labels in datasets $\{D_2, \ldots, D_{20}\}$ requires generating pseudo-labels. The accuracy of these pseudo-labels significantly impacts the model's ability to generalize across datasets. Balancing the trade-off between pseudo-label confidence and diversity remains a key challenge.

Domain Shift

Datasets $\{D_{11}, \ldots, D_{20}\}$ have distributions different from $\{D_1, \ldots, D_{10}\}$. This domain shift requires the model to adapt while preserving its performance on earlier datasets, highlighting the need for effective domain adaptation strategies.

Catastrophic Forgetting

As the model iteratively trains on new datasets, it risks forgetting knowledge from prior datasets, leading to performance degradation. Managing this requires incorporating mechanisms like prototype retention and memory-based methods.

3 Methodology

This section details the step-by-step methodology used to address the problem of iteratively updating models across multiple datasets while mitigating catastrophic forgetting and adapting to domain shifts.

Feature Extraction

We use a pre-trained ResNet-50 model for feature extraction:

- The final fully connected (classification) layer is removed to obtain a fixed-size feature vector $\mathbf{z}_i \in \mathbb{R}^{2048}$ for each input image x_i.
- The extracted features are passed to subsequent stages for prototype construction and pseudo-label generation.

Mathematically, the feature extraction process can be represented as:

$$\mathbf{z}_i = f_{\text{resnet}}(x_i; \theta_{\text{fixed}})$$

where f_{resnet} is the ResNet-50 backbone, and θ_{fixed} represents the frozen parameters of the pre-trained network.

Prototype Construction

Class prototypes $\mathbf{c}_k$ are constructed using the extracted features of labeled samples:

$$\mathbf{c}_k = \frac{1}{|D_k|} \sum_{i \in D_k} \mathbf{z}_i, \quad \forall k \in \{1, \ldots, C\}$$

where C is the number of classes and D_k is the set of feature vectors belonging to class k.

These prototypes serve as the class representatives and are pivotal for assigning pseudo-labels and aligning new data distributions with prior knowledge.

Cosine similarity for Pseudo-Labeling

For the unlabeled datasets $\{D_2, \ldots, D_{20}\}$, pseudo-labels are generated based on the similarity of each feature vector $\mathbf{z}_i$ to the class prototypes. For each unlabeled feature vector $\mathbf{z}_i$, the cosine similarity with all prototypes $\mathbf{c}_k$ is calculated.

The pseudo-label $\hat{y}_i$ is assigned to the class k_1 corresponding to the prototype with the highest similarity score:

$$\hat{y}_i = \arg\max_k \text{Sim}(\mathbf{z}_i, \mathbf{c}_k), \tag{1}$$

where the similarity is defined as:

$$\text{Sim}(\mathbf{z}_i, \mathbf{c}_k) = \frac{\mathbf{z}_i \cdot \mathbf{c}_k}{\|\mathbf{z}_i\| \cdot \|\mathbf{c}_k\|}. \tag{2}$$

High-Confidence Selection

To ensure reliable updates to prototypes:

1. Compute the difference between the highest and second-highest similarity scores (confidence).
2. Select the top fraction of samples with the highest confidence.

We chose top fraction as **0.8** .

Prototype Contrastive Alignment

To address the domain shift in datasets $\{D_{11}, \ldots, D_{20}\}$, Prototype Contrastive Alignment (PCA) is applied. This technique ensures that the feature distributions of new data align with the existing prototypes.

The implemented PCA method updates prototypes iteratively as follows:

$$\mathbf{c}_k^{\text{aligned}} = \alpha \mathbf{c}_k^{\text{prev}} + (1 - \alpha)\mathbf{c}_k^{\text{curr}}, \quad \forall k \in \mathcal{C},$$

where:

- $\mathbf{c}_k^{\text{prev}}$ is the prototype of class k from the previous iteration.
- $\mathbf{c}_k^{\text{curr}}$ is the prototype of class k calculated from the current dataset.
- $\mathbf{c}_k^{\text{aligned}}$ is the updated (aligned) prototype for class k.
- $\alpha \in [0, 1]$ is a hyperparameter controlling the balance between previous and current prototypes.
- $\mathcal{C}$ is the set of all classes present in the dataset.

If a class k exists only in the current dataset and not in the previous one, the prototype is initialized as:

$$\mathbf{c}_k^{\text{aligned}} = \mathbf{c}_k^{\text{curr}}.$$

We chose α as **0.5** .

Iterative Training and Evaluation

The same methodology was used across all unlabeled datasets D_2 to D_{20}. No distinction was made between Task 1 and Task 2; the pseudo-label generation and prototype-based approach remained consistent throughout. This uniformity ensures a systematic and reproducible pipeline for pseudo-labeling and model updates. The models are updated iteratively across datasets, with performance evaluated on both current and previously seen datasets using held-out labeled sets.

4 Observation and Analysis

Accuracy Matrix (Tabular View)

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20
Model 1	79.92	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 2	74.48	75.56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 3	74.72	75.68	76.24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 4	74.16	75.32	75.92	75.64	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 5	74.16	74.8	76.04	75.64	75.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 6	74.08	74.56	75.64	75.6	75.12	74.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 7	73.64	74.68	75.4	75.48	74.6	74.12	73.64	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 8	73.4	74.88	75.2	75.44	74.6	73.96	73.88	73.88	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 9	73.48	74.92	75.52	75.48	74.6	74.16	73.88	74.2	72.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 10	73.4	75.04	75.24	74.92	74.24	74.2	73.84	74.24	72.32	76.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 11	71.32	71.64	73.24	72.6	71.92	72.04	71.36	71.28	69.52	74.12	60.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 12	67.0	66.64	68.64	68.36	67.72	67.76	66.76	67.4	65.6	70.16	56.08	50.88	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 13	65.84	65.48	67.8	67.6	66.6	66.72	65.76	66.84	64.96	69.32	55.92	50.6	59.92	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 14	65.2	65.0	67.4	66.36	65.88	65.88	65.68	66.36	64.44	68.64	55.16	50.8	59.08	62.52	0.0	0.0	0.0	0.0	0.0	0.0
Model 15	66.68	66.8	69.04	68.44	67.4	67.56	67.0	67.6	65.52	69.8	56.2	50.8	60.72	62.96	66.84	0.0	0.0	0.0	0.0	0.0
Model 16	64.64	65.56	67.92	66.8	67.36	65.68	65.56	66.72	64.64	69.24	55.24	50.04	59.6	62.32	66.52	57.52	0.0	0.0	0.0	0.0
Model 17	64.64	64.44	67.52	66.48	66.4	65.28	65.68	66.48	64.48	68.0	55.8	50.12	59.48	62.36	65.72	57.56	58.4	0.0	0.0	0.0
Model 18	63.4	63.4	65.88	64.8	65.0	64.36	64.52	64.52	62.84	67.4	53.92	50.04	58.4	61.8	64.36	56.4	57.32	57.92	0.0	0.0
Model 19	62.8	61.96	64.56	64.56	63.88	63.44	63.2	63.6	62.32	66.72	52.44	49.44	57.92	59.68	64.2	55.64	56.08	57.2	53.36	0.0
Model 20	64.6	63.92	66.48	66.04	65.76	64.68	64.68	65.68	63.6	67.88	54.28	50.0	59.36	60.52	65.72	57.08	57.56	57.88	53.08	61.76

4.1 Accuracy Trends Across Datasets

The model's performance on datasets D_1 to D_{10} appears consistent due to their shared distribution. However, there is a noticeable drop in accuracy when moving to datasets D_{11} to D_{20}, likely caused by a domain shift (i.e., the data distribution becomes different).

4.2 Impact of Catastrophic Forgetting

As the model trains on subsequent datasets, the accuracy on earlier datasets tends to degrade. This reflects *catastrophic forgetting*, where the model overwrites its knowledge of previous datasets while learning new ones.

4.3 Domain Adaptation Challenges

The model struggles to adapt to the domain shift between D_1–D_{10} and D_{11}–D_{20}, which suggests limited generalization capabilities of the features extracted using ResNet-50 or the prototype-based learning approach.

4.4 Dataset-Specific Insights

If certain datasets (e.g., D_{12}) show consistently lower accuracy, this might indicate either greater complexity in those datasets or poor alignment with the ResNet-50 feature representations.

4.5 Overall Performance

The average accuracy across datasets provides a clear picture of the method's strengths and weaknesses. If D_1–D_{10} outperform D_{11}–D_{20} significantly, it confirms the importance of addressing domain adaptation.

5 Problem-2 (Presentation on Research Paper)

The presentation on "**Deja Vu : Continual Model Generalization for Unseen Domains (ICLR 2023)**" is available on the following link :
https://youtu.be/JX1iqUpz4Ts

6 References

- Pandas Documentation
  https://pandas.pydata.org/
- Numpy Documentation
  https://numpy.org/
- Accuracy score Documentation
  https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html
- Pytorch Documentation
  https://pytorch.org/docs/stable/index.html
  https://pytorch.org/vision/stable/
  https://pytorch.org/docs/stable/nn.html
- Pickle Documentation
  https://docs.python.org/3/library/pickle.html
- Resnet50 Documentation
  https://pytorch.org/vision/0.18/models/generated/torchvision.models.resnet50.html

  For feature extraction, we utilized a pre-trained ResNet-50 model. This neural network was trained on the ImageNet dataset, which includes over 14 million labeled images spanning 1,000 distinct categories. ResNet-50, with its deeper architecture and skip connections, is well-suited for capturing detailed feature representations. By leveraging this pre-trained model, we extracted robust features that encapsulate both low-level details and high-level semantic information, reducing the need for extensive training on our specific dataset while ensuring high-quality inputs for subsequent processing.
- Deja Vu : Continual Model Generalization for Unseen Domains
  https://arxiv.org/pdf/2301.10418