CS771 - Intro to ML (Autumn 2024): Mini-project 2

November 26, 2024

Sagili Harshitha - 220934 Sarath Kumar - 220754 Bhanu Prakash - 220583
Sree Sahithi - 220577 Dinesh Naik - 220866

1 Introduction

1.1 Problem Statement

This project involves 20 training datasets (D_1 to D_{20}) derived from the CIFAR-10 image classification dataset. The first 10 datasets share a common input distribution, while the remaining 10 have varying but similar distributions. Only D_1 is labeled; the rest are unlabeled. Additionally, 20 labeled held-out datasets ($\hat{D}_1$ to $\hat{D}_{20}$) are provided for model evaluation.

1.1.1 Task-1

Task 1 aims to trains models $f_1, f_2, \ldots, f_{10}$ using datasets D_1 to D_{10}. f_1 is trained on the labeled dataset D_1 with an LwP classifier, then used to generate pseudo-labels for D_2 to train f_2. This process is repeated iteratively, producing models f_1 to f_{10}. Each model f_i is evaluated on its held-out dataset $\hat{D}_i$ and all prior held-out datasets $(\hat{D}_1, \ldots, \hat{D}_{i-1})$, resulting in a 10×10 accuracy matrix to assess task performance and knowledge retention.

1.1.2 Task-2

Task 2 builds on Task 1 by training models $f_{11}, f_{12}, \ldots, f_{20}$ on datasets D_{11} to D_{20} with varying input distributions. Starting from f_{10}, f_{11} is trained on D_{11}, and its predictions pseudo-label D_{12} to train f_{12}. This process continues until f_{20} is trained. Each model f_i is evaluated on its held-out dataset $\hat{D}_i$ and all prior ones $(\hat{D}_1, \ldots, \hat{D}_{i-1})$, creating a 10×20 accuracy matrix to assess adaptation and retention.

1.2 Aim

The problem aims to train models that maintain high accuracy across new and previously seen datasets, minimizing performance degradation as new domains are introduced. The focus is on addressing continual learning challenges while reducing catastrophic forgetting, even with varying input distributions.

2 Dataset Description

The dataset includes 20 training and 20 evaluation datasets, divided into two tasks.

Task-1 Training Data (D_1 to D_{10})

Task-1 training data consists of 10 datasets:

- (D_1): Labeled dataset with 32×32 RGB images and targets $\{0, 1, 2, \ldots, 9\}$.

- **(D_2) to (D_{10})**: Unlabeled datasets with 32×32 RGB images, requiring pseudo-labeling for training.

Task-2 Training Data (D_{11} to D_{20})

Task-2 training data includes datasets D_{11} to D_{20}, which are unlabeled and require pseudo-labeling using predictions from models trained on earlier datasets.

- **(D_{11} to D_{20})**: 32×32 RGB images, with no labels (pseudo-labeling needed).

Key Properties

- **Data Representation and Labeling**: Each input is a 32×32 RGB image from CIFAR-10. Datasets D_{11} to D_{20} differ in input distributions. Training datasets (D_2 to D_{20}) require pseudo-labeling, while evaluation datasets are fully labeled.
- **Workflow**:
  - Train f_1 on labeled D_1.
  - Use f_1 to pseudo-label D_2 and train f_2.
  - Repeat iteratively up to D_{20}, training f_{20}.
  - Evaluate each f_i on $\hat{D}_i$ and earlier held-out datasets to monitor performance.

3 Analysis

Task 1: Training and Evaluation on Uniform Distribution Datasets

Flattening-Approach

- **Feature Extraction**: Resize images to 100×100 and flatten into 1D vectors.
- **Classifier (LwP)**: Compute mean feature vectors as class prototypes and classify using Euclidean distance.
- **Training**:
  - f_1: Train on D_1 with true labels.
  - f_2 to f_{10}: Use f_{i-1} to pseudo-label D_i and update prototypes.
- **Evaluation**: Evaluate f_i on D_1 to D_i and record validation accuracy.

ResNet-Approach

Feature Extraction with ResNet

- A pre-trained ResNet-152 model is used for feature extraction.

- Images are resized to 160×160 and pre-processed using normalization with ImageNet mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$.
- Deep features are extracted from the penultimate layer of ResNet-152, resulting in a compact embedding for each image.
- The feature extractor operates in evaluation mode to prevent weight updates during inference.

Classifier: Lightweight Prototype Classifier (LwP)

- Computes class prototypes as the mean feature vector for each class.
- Classifies samples based on the nearest class prototype using Euclidean distance.

Training Process

- **Model** f_1: Trained directly on D_1 using true labels.
- **Models** $f_2, \ldots, f_{10}$:
  - Predict pseudo-labels for D_i using the previous model f_{i-1}.
  - Update the prototypes with the new dataset D_i using its pseudo-labels.

Evaluation

- Each model f_i is evaluated on datasets $\hat{D}_1$ to $\hat{D}_i$
- Performance is recorded as validation accuracy for each dataset.

Task 2: Training and Evaluation on Non-Uniform Distribution Datasets

Flattening-Approach

- **Feature Extraction**: Images are resized to 100×100 and flattened into feature vectors, as in Task 1.
- **Classifier Initialization**: Start with f_{10} (final model from Task 1) and update it sequentially using D_{i+10}.
- **Training**:
  - Predict pseudo-labels for D_{i+10} using f_i.
  - Update prototypes with pseudo-labeled D_{i+10}.
- **Evaluation**: Evaluate f_i on $D_{11}, D_{12}, \ldots, D_i$, recording validation accuracy.

ResNet-Approach

Feature Extraction with ResNet

- We used the same feature extraction method for Task 2 as we did for Task 1.

Classifier: Lightweight Prototype Classifier (LwP)

- We used the same LwP classifier for Task 2.
- Begin with f_{10} (final model from Task 1) as the initial model for Task 2.
- Sequentially update the classifier using new datasets D_{i+10}.

Training Process

- Predict pseudo-labels for D_{i+10} using the previous model f_{i+9}.
- Update the prototypes with the new dataset D_{i+10} using its pseudo-labels.

Evaluation

- Each model f_{i+10} is evaluated on datasets $\hat{D}_1$ to $\hat{D}_{i+10}$
- Validation accuracy is recorded for each dataset.

Conclusion

1. Best Feature Extraction Approach

The second feature extraction approach using ResNet-152 with embeddings from the penultimate layer clearly outperforms the flattened feature extraction approach. This is evident from the significantly higher accuracy values in the accuracy matrix for both Task-1 and Task-2. The deep ResNet-152 features effectively capture more complex and meaningful representations, leading to improved classification performance.

2. Accuracy Analysis for ResNet-152 Feature Extraction

- **Task-1 (Model Index [1,10]):**
  Models trained on Task-1 exhibit consistent high accuracy values ranging from **0.88 to 0.91** across datasets, indicating robust performance. The ResNet-152 embeddings enable the lightweight prototype classifier to generalize well within the Task-1 datasets.

- **Task-2 (Model Index [11,20]):**
  Models trained on Task-2 demonstrate a slight drop in accuracy compared to Task-1, with values ranging between **0.72 and 0.86**, except for datasets 12 and 19, which achieve accuracies of **0.53** and **0.59**, respectively. However, the approach still maintains competitive performance on later datasets, showcasing its ability to adapt and update class prototypes effectively when incorporating new data.

- Transforming images to higher dimensions, such as **above 200**, results in **overfitting** of the models. Conversely, reducing the dimensions to **less than 100** causes the models to **underfit**, making them unable to learn the features effectively.

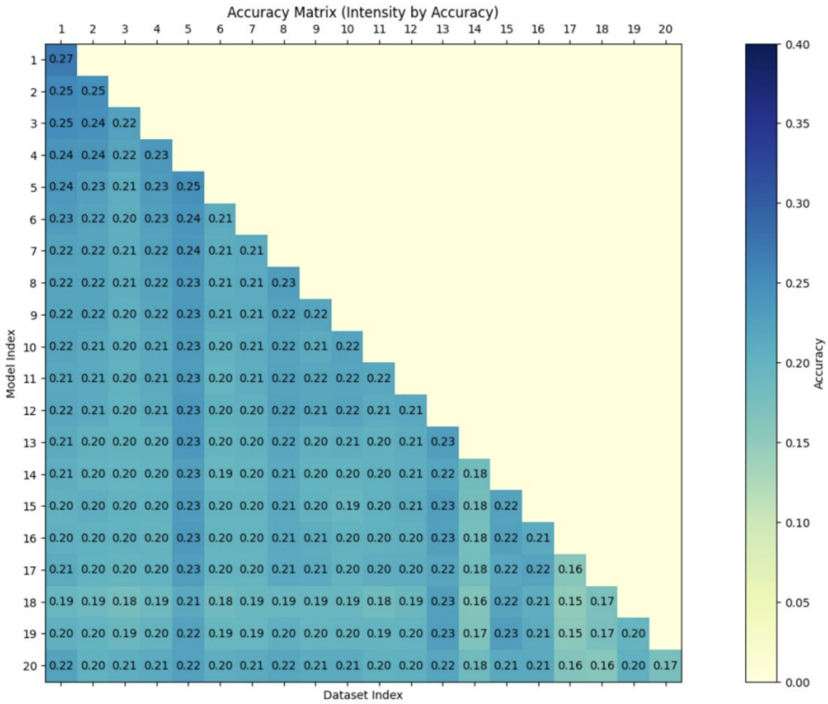

Figure 1: With Flattened feature extraction

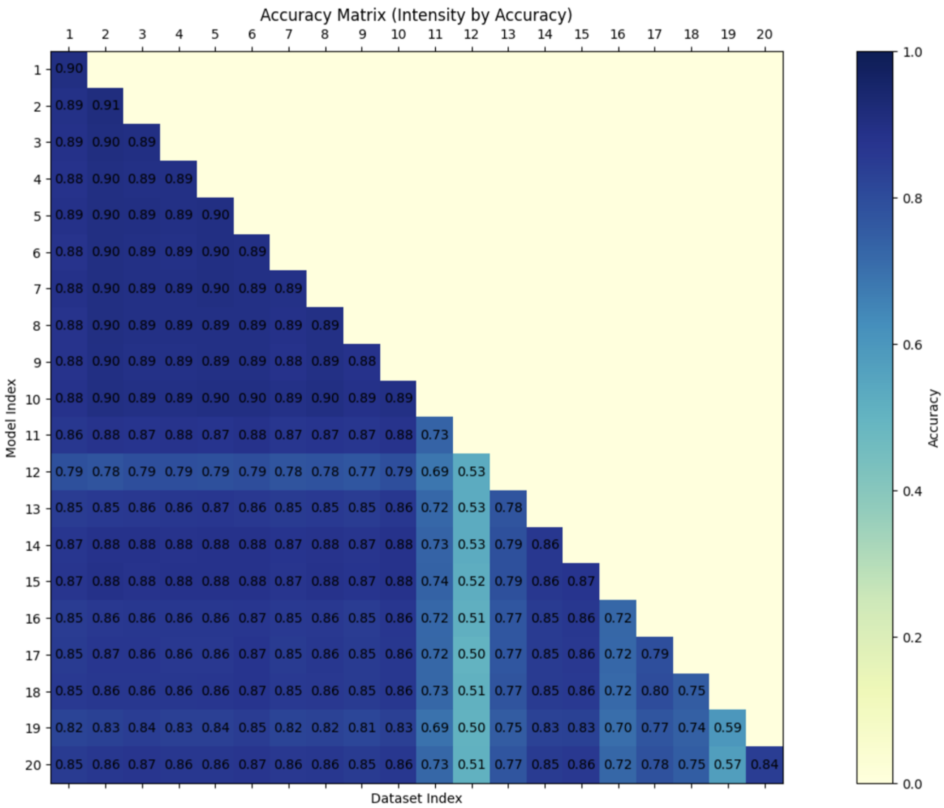

Figure 2: With ResNet feature extraction

Overall, the ResNet-152-based feature extraction method with **160 × 160** transformed images provides a clear advantage in terms of accuracy and adaptability for both **Task-1** and **Task-2**.

4 References:

- For Sklearn: https://scikit-learn.org/stable/index.html
- For Tensorflow: https://www.tensorflow.org/ and TensorFlow YouTube Playlist
- For Python and its libraries: https://www.w3schools.com/python/
- To know more about feature extractions we used: Feature Extraction Paper

5 Problem 2

- Video link: https://https://www.youtube.com/watch?v=Zca7tXp8Q1c

Acknowledgement

We would like to express our heartfelt gratitude to our professor, Dr. Piyush Rai, for his invaluable guidance and support throughout this mini-project. His expertise and encouragement were instrumental in our successful exploration and implementation of incremental learning tasks using the LwP classifier. We are deeply thankful for the opportunity to work on these challenging tasks, which allowed us to gain hands-on experience with advanced machine learning techniques and address practical challenges like distributional shifts across datasets. This experience has significantly deepened our understanding of incremental learning and enhanced our problem-solving and collaboration skills. Thank you, Dr. Rai, for your mentorship and for fostering a stimulating and supportive learning environment.