CS771 : Mini-Project 2

Group No: 72

Venkatesh
220109

Manikanta
220409

Prashant
220803

Sai Nikhil
221095

Pankaj Nath
221188

1 Introduction

This mini-project focuses on developing and evaluating continual learning models across multiple datasets with varying input distributions. Our goal is to implement a Learning with Prototype-based (LwP) classifier that can effectively adapt to new datasets while maintaining performance on previously learned tasks.

1.1 Project Objectives

The project consists of two main tasks:

Task-1: Continual Learning with Identical Input Distributions

Initially, a prototype-based model, labeled (f_1), is trained on a starting dataset (D_1). This model is subsequently refined over several iterations (f_1 to f_{10}) leveraging predictions on new datasets (D_1 to D_{10}) with similar distributions. Each updated model is evaluated on the corresponding held-out datasets ($\hat{D}_1$ to $\hat{D}_{10}$), and the results are summarized in a performance matrix. The main objective is to ensure consistent performance on newly added data sets while preserving accuracy on earlier ones.

Task-2: Continual Learning with Varied Input Distributions

Building on the final model from Task 1 (f_{10}), this phase introduces data sets (D_{11} to D_{20}) with different input distributions to further refine the model (f_{11} to f_{20}). Evaluations are performed in all held-out datasets ($\hat{D}_1$ to $\hat{D}_{20}$) to measure the ability of the model to generalize to new distributions while maintaining its effectiveness on previously encountered data.

The project emphasizes achieving a balance between learning new tasks and retaining performance on previous ones, addressing the dual challenges of adaptability and memory retention.

2 Problem 1: Continual Learning with LwP

2.1 Dataset Description

The project involves working with 20 datasets derived from the CIFAR-10 image classification dataset. These datasets are unique in their input distributions and can be categorized into two distinct groups:

2.2 First Set of Datasets: Consistent Input Distribution

2.2.1 Data Characteristics

- Datasets: $D_1, D_2, \ldots, D_{10}$
- Input Distribution: Consistent across these 10 datasets
- Feature Representation: Raw 32×32 color images
- Labeling: Only D_1 is initially labeled

2.3 Second Set of Datasets: Varied Input Distributions

2.3.1 Data Characteristics

- Datasets: $D_{11}, D_{12}, \ldots, D_{20}$
- Input Distribution: Potentially different for each dataset
- Similarity: Some degree of similarity with the first group's distribution
- Feature Representation: Raw 32×32 color images

2.4 Approach Overview

In this problem, we will solve a continual learning task using the LwP classifier across two distinct scenarios:

- **Task 1:** Learning from 10 datasets with the same input distribution
- **Task 2:** Learning from 10 datasets with potentially different input distributions

2.5 Methodology

2.5.1 Feature Extraction

For feature extraction, we use a probabilistic classifier that leverages the Gaussian Mixture Model (GMM) for capturing class-specific distributions. The feature representation technique involves extracting features from raw images and applying preprocessing steps to standardize and normalize the data. This is the same for both Task 1 and Task 2. Here are the specific details:

- **Feature Extraction Method:** We first use raw images of size 32×32 as input. The features are then scaled using a standard scaler to ensure consistency across different datasets and to facilitate better model performance.
- **Pre-processing Steps:** The input images are scaled using a standardization technique where each feature (pixel) is normalized to have zero mean and unit variance. This is performed using the `StandardScaler` from the `sklearn` library.
- **Neural Network Used for Feature Representation:** While a pre-trained neural network could be used for feature extraction, for simplicity and focus on the probabilistic classifier, raw image features are processed directly using a scaling method. We employed models such as ResNet pre-trained on ImageNet for initial feature extraction, adapting them to the CIFAR-10 dataset specifics.

2.5.2 Model Strategy

The model strategy revolves around sequentially updating the classifier with new datasets using the predictions from previous models. We used the Gaussian Mixture Model approach to fit a multivariate normal distribution to the data for each class and then make predictions based on the log-probability of the samples under those distributions. The process for model updates is explained below:

Task 1.1: Initial Model Training and Iterative Updates

1. **Initial Model Training**:
   - The first dataset D_1 (labeled) is used to train the initial model f_1.

- Features are scaled using `StandardScaler`, and class prototypes, covariance matrices, and priors are computed.
- Multivariate Gaussian distributions are used for probabilistic predictions.

2. **Iterative Updates**:
   - For each subsequent dataset D_i ($i = 2, \ldots, 10$):
     - Predict pseudo-labels for D_i using the current model f_{i-1}.
     - Filter predictions based on confidence levels (e.g., $> 90\%$ confidence).
     - Use the filtered pseudo-labeled data to update the prototypes, covariances, and priors of the LwP model.
     - Ensure that updated parameters blend the existing knowledge with the new dataset to avoid catastrophic forgetting.

Task 1.2: Domain Adaptation and Iterative Refinement

1. **Initialization**:
   - The final model f_{10} from Task 1.1 serves as the starting point for Task 1.2.
2. **Domain Adaptation**:
   - Pseudo-datasets are generated using the Gaussian distributions defined by the prototypes, covariances, and priors.
   - The sliced Wasserstein distance (SWD) is computed between the target dataset and the pseudo-dataset.
   - Updates prioritize reducing SWD, ensuring alignment between the current dataset and the learned distribution.
3. **Iterative Refinement**:
   - For each dataset D_i ($i = 11, \ldots, 20$):
     - Perform iterative updates to minimize SWD while maintaining model stability.
     - Adapt prototypes, covariances, and priors using confident samples from the target dataset.

2.6 Performance Analysis

2.6.1 Task 1 Results

We present the accuracy matrix for the first 10 models across the first 10 held-out datasets. This matrix shows how each model f_i performs on the held-out datasets $\hat{D}_1$ to $\hat{D}_{10}$. The accuracy values are computed by applying each model on all previous held-out datasets, ensuring that the performance of earlier datasets does not degrade with the update of each new model. The table below summarizes the performance of each model on the held-out datasets.

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	0.8760	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_2	0.8780	0.8904	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_3	0.8772	0.8900	0.8724	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_4	0.8780	0.8908	0.8724	0.8808	0.0	0.0	0.0	0.0	0.0	0.0
f_5	0.8780	0.8900	0.8728	0.8808	0.8760	0.0	0.0	0.0	0.0	0.0
f_6	0.8784	0.8900	0.8732	0.8808	0.8768	0.8816	0.0	0.0	0.0	0.0
f_7	0.8788	0.8892	0.8724	0.8796	0.8752	0.8820	0.8720	0.0	0.0	0.0
f_8	0.8784	0.8900	0.8736	0.8784	0.8756	0.8832	0.8696	0.8836	0.0	0.0
f_9	0.8776	0.8900	0.8736	0.8772	0.8752	0.8824	0.8688	0.8840	0.8756	0.0
f_{10}	0.8768	0.8916	0.8744	0.8772	0.8764	0.8816	0.8684	0.8852	0.8740	0.8808

Table 1: Accuracy Matrix for Task 1

2.6.2 Task 2 Results

Model	$\hat{D}_{11}$	$\hat{D}_{12}$	$\hat{D}_{13}$	$\hat{D}_{14}$	$\hat{D}_{15}$	$\hat{D}_{16}$	$\hat{D}_{17}$	$\hat{D}_{18}$	$\hat{D}_{19}$	$\hat{D}_{20}$
f_{11}	0.7184	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_{12}	0.7184	0.5312	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_{13}	0.7184	0.5312	0.7492	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_{14}	0.7184	0.5308	0.748	0.7636	0.0	0.0	0.0	0.0	0.0	0.0
f_{15}	0.7184	0.5308	0.748	0.7636	0.8696	0.0	0.0	0.0	0.0	0.0
f_{16}	0.7184	0.5308	0.748	0.7636	0.8696	0.7148	0.0	0.0	0.0	0.0
f_{17}	0.7184	0.5308	0.748	0.7636	0.8696	0.7148	0.724	0.0	0.0	0.0
f_{18}	0.7184	0.5308	0.748	0.7636	0.8696	0.7148	0.724	0.7156	0.0	0.0
f_{19}	0.7184	0.5308	0.7484	0.7636	0.8696	0.7148	0.7244	0.7156	0.698	0.0
f_{20}	0.7184	0.5308	0.748	0.7636	0.8696	0.7148	0.7244	0.7156	0.6976	0.8156

Table 2: Accuracy Matrix for Task 2

2.7 Key Observations

- Until unless specifically handled, the model forgets about the old distribution while learning about the new distributions.
- Pseudo-datasets are generated from the generative models that are captured during training each dataset and are replayed while training on further datasets.
- The update strategy used for the model effectively mitigated the catastrophic forgetting and yielded almost the same accuracies in later runs also.

3 Problem 2: Paper Presentation

- Selected Paper: Lifelong Domain Adaptation via Consolidated Internal Distribution (NeurIPS 2021)
- Presentation Link: YouTube video
- Slides used in video: Presentation

3.1 Summary of Presentation

- Task: Adapting Models Across Domains
- Limitations of Unsupervised Domain Adaptation (UDA) and Continual Learning (CL)
- Proposed Solution: LDAuCID
- Experimental Results:
  - Adaptation: Improves performance on target domains.
  - Retention: minimal forgetting of past tasks.
  - Competitive Results: matches or outperforms state-of-the-art UDA methods.