Mini-Project-2

CS771 Introduction to Machine Learning

Prof. Piyush Rai

Subham Anand (221093) Rythm Kumar (220925) Viral Chitlangia (221198)
Shreyas Dwivedi (221035) Arunkumar Duraisamy (220211)

1 Introduction

Problem 1 involves sequential model training and evaluation across two tasks.

In **Task 1**, we are given 10 CIFAR-10 subsets, with only the first dataset labeled. The goal is to train models sequentially, ensuring good performance on both the current and previously seen datasets.

In **Task 2**, we are provided with 10 additional unlabeled datasets with distributional shifts. The objective is to continue training on these new datasets, starting from the model trained in Task 1, while adapting to the distribution changes.

2 Problem 1: CIFAR-10 Description

The CIFAR-10 dataset is a widely used benchmark for image classification tasks. It consists of 60,000 color images, each of size 32×32 pixels, across 10 distinct classes. The small image size and varied data distribution make it a challenging dataset for machine learning models.

2.1 Training Datasets

- $D_1, D_2, \ldots, D_{20}$: We are given 20 subsets of CIFAR-10.
- D_1 to D_{10}: These datasets share the same input distribution $p(x)$.
- D_{11} to D_{20}: These datasets have unique input distributions that are similar to $p(x)$, but not identical.

2.2 Evaluation Datasets

- $D'_1, D'_2, \ldots, D'_{20}$: These are 20 held-out labeled subsets for model evaluation.

3 Feature Extraction

3.1 Model Used: DenseNet121

DenseNet121 is a CNN with a densely connected architecture, allowing efficient feature reuse and improved gradient flow, making it ideal for smaller datasets like CIFAR-10.

3.2 Pretrained Weights: ImageNet

The model is initialized with weights pretrained on ImageNet, providing generalized feature representations learned from over a million images across 1,000 classes.

3.3 Modification: Removal of Classification Layer

The original classification layer is removed, repurposing the model for feature extraction. The modified DenseNet121 outputs high-dimensional feature vectors for class prototypes and covariance matrix calculation.

4 Task 1: Sequential Training

4.1 Problem Description

For each i-th dataset D_i, we train the model on D_i and then evaluate it on all previous evaluation datasets D'_k, where $k \leq i$.

4.2 Approach

1. **Training on** D_1:
   - Begin by training the model on the labeled dataset D_1.
   - Compute class prototypes and the covariance matrix using the Mahalanobis distance.
   - Evaluate on D'_1 with an accuracy of 85%.
2. **Training on** D_2 **to** D_{10}:
   - For each subsequent dataset D_i (where $i > 1$), the model f_{i-1}, trained on D_{i-1}, is used to predict labels for the training dataset D_i.
   - Generate 100 pseudo-data points per class to represent the previous datasets $D_1, D_2, \ldots, D_{i-1}$, inspired by the paper *Lifelong Domain Adaptation via Consolidated Internal Distribution*.
   - Note: These are not actual data points from previous datasets, but representative pseudo-data.
   - Use these pseudo-data points and the predicted labels from D_i to train a new model f_i.
   - Compute the new class prototypes and covariance matrix.
   - Test f_i on the evaluation dataset D'_i.
   - Repeat the process for all datasets $D_3, D_4, \ldots, D_{10}$.

$$\begin{pmatrix}
85.72 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
82.44 & 83.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
81.36 & 81.08 & 81.04 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
80.20 & 79.48 & 79.72 & 80.08 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
77.40 & 79.00 & 77.44 & 79.00 & 78.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
77.44 & 77.44 & 77.68 & 77.00 & 76.80 & 78.24 & 0.00 & 0.00 & 0.00 & 0.00 \\
76.52 & 76.68 & 75.68 & 76.72 & 75.88 & 77.20 & 75.64 & 0.00 & 0.00 & 0.00 \\
73.76 & 74.44 & 73.60 & 74.40 & 74.12 & 74.44 & 73.72 & 73.24 & 0.00 & 0.00 \\
73.72 & 74.68 & 72.60 & 74.28 & 73.88 & 74.04 & 73.20 & 72.80 & 70.84 & 0.00 \\
72.24 & 74.40 & 72.36 & 73.16 & 73.84 & 73.28 & 72.88 & 71.08 & 69.92 & 73.92
\end{pmatrix}$$

Table 1: Evaluation Matrix For Task 1

4.3 Evaluation

The accuracy of the models is reported in a 10×10 matrix:

- **Rows**: Evaluation datasets $D'_1, D'_2, \ldots, D'_{10}$.
- **Columns**: The i-th iteration where the model was updated using D_i.

5 Task 2: Adapting to Drifted Distributions

5.1 Problem Description

In **Task 2**, we continue training from the model f_{10} obtained from Task 1. Now, we are given 10 additional datasets $D_{11}, D_{12}, \ldots, D_{20}$, each with drifted distributions compared to the earlier datasets. These datasets are unlabeled. The goal is to adapt the model to these new distributions and continue training sequentially.

5.2 Approach

1. **Training on** D_{11}:
   - Use the model f_{10}, trained on the earlier datasets, to predict labels for the training dataset D_{11}.
   - Generate 100 pseudo-data points per class representing earlier datasets $D_1, D_2, \ldots, D_{10}$.
   - Update the covariance matrix to adapt to the new distribution of D_{11}.
   - Train a new model f_{11} using the pseudo-data points and the predicted labels from D_{11}.
   - Test f_{11} on the evaluation dataset D'_{11}.
2. **Training on** D_{12} **to** D_{20}:
   - Repeat the process for each subsequent dataset $D_{12}, D_{13}, \ldots, D_{20}$, using the model from the previous iteration to predict labels and generate pseudo-data points.
   - Evaluate on all evaluation datasets D'_k for $k \leq i$, where i is the current iteration.

5.3 Evaluation

The accuracy of the models is reported in a 10×20 matrix:

- **Rows**: Evaluation datasets $D'_1, D'_2, \ldots, D'_{20}$.
- **Columns**: The i-th iteration where the model was updated using D_i.

Evaluation Accuracy Matrix:

	0	1	2	3	4	5	6	7	8	9	\
0	63.84	65.72	63.92	64.64	65.40	64.68	65.68	64.16	62.64	65.24	
1	51.28	52.68	53.44	53.60	54.04	52.68	53.00	51.92	51.16	53.92	
2	55.92	55.96	57.28	55.76	58.16	55.96	56.00	54.72	53.08	56.48	
3	58.72	58.40	57.44	57.04	59.84	57.96	58.12	57.16	55.76	58.52	
4	58.32	61.76	59.16	59.24	60.80	59.88	58.80	58.72	56.96	59.40	
5	53.44	56.40	54.56	53.68	56.08	53.88	53.56	53.68	52.32	54.04	
6	55.44	56.64	56.12	55.52	57.48	56.72	55.00	54.52	53.84	55.88	
7	54.60	57.16	54.00	52.40	57.20	55.20	54.44	54.52	54.00	54.96	
8	48.12	48.52	48.44	48.48	50.36	48.36	48.80	48.68	46.24	51.24	
9	50.68	52.64	51.80	50.88	52.68	50.16	49.56	50.32	48.80	51.96	

	10	11	12	13	14	15	16	17	18	19
0	46.80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	38.32	29.52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	38.80	29.24	41.92	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	39.96	27.88	44.32	54.28	0.00	0.00	0.00	0.00	0.00	0.00
4	37.80	27.48	44.88	54.40	58.88	0.00	0.00	0.00	0.00	0.00
5	32.32	25.88	44.08	46.48	52.88	41.68	0.00	0.00	0.00	0.00
6	34.80	23.44	42.68	50.20	55.08	39.64	48.08	0.00	0.00	0.00
7	33.68	22.08	42.52	50.12	53.84	40.08	47.16	42.08	0.00	0.00
8	29.64	23.36	39.12	45.88	48.48	36.68	41.92	39.08	32.40	0.00
9	31.88	24.16	40.16	46.68	51.56	34.44	42.80	41.04	33.48	47.24

Figure 1: Evaluation Accuracy For Task 2

6 Problem 2

We were given two papers to choose from:

1. *Lifelong Domain Adaptation via Consolidated Internal Distribution* (Link to the paper)

2. *Deja vu: Continual Model Generalization for Unseen Domains* (Link to the paper)

Out of these, we selected **Research Paper 1** (*Lifelong Domain Adaptation via Consolidated Internal Distribution*).

Here is the YouTube link explaining our understanding of the Paper 1. (Slides)

7 Conclusion

- The evaluation accuracy remains stable across models 1 to 10, with consistent performance on both current and previous datasets. Accuracy for model i is calculated on all evaluation datasets D'_k where $k \leq i$, showing no significant drop as the models progress. This indicates effective learning and retention of previously learned distributions throughout the training process.

- The evaluation accuracy shows a gradual decrease from models 11 to 20 across datasets 1 to 20. However, the models seem relatively stable, with accuracy on previous datasets not dropping rapidly. and the overall trend indicates consistent performance despite the drifted distributions in the later datasets.