CS771 Mini-Project 2

Group No. 47: Log 1 Base 1

Gottupulla Venkata Aman 220413

K S U Rithwin 220537

Mitesh Prashant Wandhare 220650

Mohammed Anas 220654

Taneshq Zendey 221123

November 26, 2024

Problem 1

Dataset Overview

The dataset comprises of subsets D1 to D20 derived from the CIFAR-10 image classification dataset. The inputs share the same distribution and have similar feature characteristics. Each subset contains raw 32×32 color images. Labels are provided only for D1, while the remaining 19 subsets are unlabeled.

1.1.1 Final Model

The model is designed to classify images efficiently by combining feature extraction, an LwP classifier, and iterative learning methods. The process involves several key steps:

1. Feature Extraction: The first stage employs a pre-trained ResNet34 model trained on ImageNet as a feature extractor. By removing its final fully connected layer, the model focuses solely on generating feature embeddings from the input images.
The images are resized to 224×224, normalized using standard mean and standard deviation values, and converted to tensors. These tensors are then passed through the truncated ResNet34 to produce a compact 512-dimensional feature vector for each image. This step condenses high-dimensional image data into informative representations without requiring the model to be trained from scratch, leveraging the power of pre-trained models.

2. Classifier Design: The LwP Classifier uses a simple yet effective design. It maintains a prototype computed during training, which is the mean feature vector for each class.
The feature vectors are normalized to ensure uniform scaling. The classifier calculates the cosine similarity between the input feature vector and each class prototype for prediction. The class with the highest similarity is predicted as the label. Additionally, probabilistic predictions are generated by applying a softmax transformation to the similarity scores.

3. Iterative Model Updating: To improve performance, the model incorporates an iterative learning approach using self-training. The classifier generates predictions (pseudo-labels) for unlabeled data in this step.
Predictions with confidence scores exceeding a predefined threshold are selected for further training. These high-confidence samples are weighted by their confidence levels, ensuring that the most reliable pseudo-labels have a greater influence on the model's updates.
This iterative process allows the model to leverage unlabeled data effectively.

4. Same Distribution Datasets: In this task, the model is trained and evaluated on ten datasets (D1 to D10) with the same data distribution. The process begins with training the model on a labeled dataset, D_1, using the extracted features and true labels. For subsequent datasets (D_2 to D_{10}), the features are extracted, and pseudo-labels are generated using the existing model.

The model is then updated iteratively using confident predictions. After each update, the model's performance is evaluated on all previously seen datasets.

This evaluation is recorded in an accuracy matrix, which tracks the model's ability to retain knowledge while learning from new datasets.

5. Task 2: Different Distribution Datasets: In this task, the model adapts to ten new datasets (D_{11} to D_{20}) with different data distributions. A domain adaptation mechanism is added to the learning process.

The domain confidence score is calculated based on the similarity between the features of the new data and the existing class prototypes. This score, combined with the model's confidence in its predictions, determines the weight of each sample during training.

The model adjusts more effectively to the new distributions by incorporating domain-specific information. The updated model is evaluated on both previously seen datasets (D1 to D10) and the new datasets, with results captured in an extended accuracy matrix.

6. Accuracy Tracking: The model's performance is systematically recorded in accuracy matrices. In Task 1, the rows of the matrix represent models trained sequentially on datasets D_1 to D_{10}, while the columns correspond to evaluations on these datasets.

In Task 2, the matrix extends to include evaluations on datasets D_{11} to D_{20}. This structure clearly visualizes the model's performance across datasets, highlighting its adaptability and robustness.

The accuracy matrices provide a detailed understanding of the model's learning and adaptability across diverse scenarios.

Task 1 Accuracy Matrix:

$$\begin{bmatrix}
0.8152 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.7916 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.7752 & 0.7916 & 0.7932 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.774 & 0.7816 & 0.7896 & 0.7828 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.7744 & 0.7908 & 0.784 & 0.788 & 0.7848 & 0 & 0 & 0 & 0 & 0 \\
0.7656 & 0.7808 & 0.7788 & 0.78 & 0.7756 & 0.778 & 0 & 0 & 0 & 0 \\
0.7684 & 0.7768 & 0.778 & 0.778 & 0.7788 & 0.776 & 0.7728 & 0 & 0 & 0 \\
0.7636 & 0.7764 & 0.7736 & 0.774 & 0.78 & 0.7748 & 0.7636 & 0.7648 & 0 & 0 \\
0.762 & 0.772 & 0.7788 & 0.7796 & 0.7796 & 0.7764 & 0.77 & 0.7664 & 0.7576 & 0 \\
0.7572 & 0.7744 & 0.776 & 0.7764 & 0.7752 & 0.774 & 0.7688 & 0.7644 & 0.7596 & 0.7868
\end{bmatrix}$$

Task 2 Accuracy Matrix:

$$\begin{bmatrix}
0.628 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.4924 & 0.4756 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5292 & 0.446 & 0.5604 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5628 & 0.4524 & 0.5744 & 0.6208 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5932 & 0.4604 & 0.6104 & 0.6484 & 0.6752 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5604 & 0.4484 & 0.5968 & 0.6312 & 0.6516 & 0.5844 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5908 & 0.4584 & 0.5944 & 0.6548 & 0.6488 & 0.562 & 0.6328 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5736 & 0.458 & 0.5916 & 0.648 & 0.6436 & 0.5476 & 0.6152 & 0.5988 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5336 & 0.4304 & 0.57 & 0.606 & 0.636 & 0.5388 & 0.5884 & 0.5728 & 0.5296 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5772 & 0.4476 & 0.6204 & 0.6452 & 0.6808 & 0.6036 & 0.6244 & 0.6152 & 0.492 & 0.6616 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

In summary, this pipeline integrates pre-trained feature extraction, a simple yet powerful classification mechanism, and iterative learning with domain adaptation. It is designed to handle both same-distribution and different-distribution datasets efficiently, ensuring strong performance while being computationally lightweight.

Problem 2

Link to the Youtube video : https://www.youtube.com/watch?v=FehRblIjfXA