Mini Project 2
Group-57

Ayushmaan Jay Singh
220276

Mayank Gupta
220638

Nikhil Gupta
220708

Riya Mittal
220901

Tamoghna Kumar
221121

Task 1.1: Iterative Training on Similar Distributions

Objective

The task involves training models $f_1, f_2, \ldots, f_{10}$ sequentially using datasets $D_1, D_2, \ldots, D_{10}$, which originate from the same input distribution. The goal is to achieve high accuracy on newly introduced datasets while ensuring performance on prior datasets does not degrade significantly.

Approach Overview

We utilized MobileNet for feature extraction, PCA for dimensionality reduction, and a custom distillation-based classifier to iteratively train models.

Methodology

1. Feature Extraction with MobileNet

MobileNet, pre-trained on ImageNet, was used to extract high-level semantic features.

- **Preprocessing**: Resized images to 224×224 and normalized using MobileNet preprocessing functions.
- **Feature Extraction**: Features were extracted from the global average pooling layer.

2. Dimensionality Reduction with PCA

PCA was applied to reduce feature vectors to 256 dimensions, retaining most of the variance for computational efficiency. Reduced features were cached for reuse.

3. Classification Using Distillation-Based Classifier

The classifier used prototype-based learning:

- **Prototype Learning**: Prototypes were initialized from labeled data in D_1.
- **Updating Mechanism**: Prototypes were iteratively refined using pseudo-labeled data from subsequent datasets.
- **Prediction**: Labels were assigned based on the nearest prototype in feature space.

4. Iterative Training and Evaluation

- Model f_1 was trained on D_1, which had labeled data.
- Pseudo-labeled data from $D_2, \ldots, D_{10}$ were used to iteratively update $f_2, \ldots, f_{10}$.
- Models were evaluated on all held-out datasets $D_1, \ldots, D_{10}$.

Results

The accuracy matrix for Task 1.1 is shown below:

```
Accuracy matrix:
[[0.7548 0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.7548 0.7424 0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.7548 0.7424 0.7372 0.     0.     0.     0.     0.     0.     0.    ]
 [0.7548 0.7424 0.7372 0.7516 0.     0.     0.     0.     0.     0.    ]
 [0.7548 0.7424 0.7372 0.7516 0.7532 0.     0.     0.     0.     0.    ]
 [0.7548 0.7424 0.7372 0.7516 0.7532 0.7528 0.     0.     0.     0.    ]
 [0.7548 0.7424 0.7372 0.7516 0.7532 0.7528 0.734  0.     0.     0.    ]
 [0.7548 0.7424 0.7372 0.7516 0.7532 0.7528 0.734  0.7548 0.     0.    ]
 [0.7548 0.7424 0.7372 0.7516 0.7532 0.7528 0.734  0.7548 0.7264 0.    ]
 [0.7548 0.7424 0.7372 0.7516 0.7532 0.7528 0.734  0.7548 0.7264 0.7596]]
```

Figure 1: Accuracy matrix for Task 1.1 showing performance of models $f_1, \ldots, f_{10}$ on held-out datasets $D_1, \ldots, D_{10}$.

—

Task 1.2: Iterative Training on Varying Distributions

This can be used for 1.1 also and gives better accuracies. So we have calculated f10 of 1.1 using this algorithm to be fed into f11 and so on.

Objective

This task extends the iterative training approach to datasets $D_{11}, D_{12}, \ldots, D_{20}$, which originate from varying input distributions. The goal is to maintain robustness to distributional shifts while ensuring strong performance on each dataset.

Approach Overview

We adopted the same MobileNet-PCA-feature distillation pipeline as in Task 1.1, with modifications to account for varying input distributions.

Methodology

1. Feature Extraction with MobileNet

MobileNet, pre-trained on ImageNet, was used to extract high-level semantic features.

- **Preprocessing**: Resized images to 224×224 and normalized using MobileNet preprocessing functions.
- **Feature Extraction**: Features were extracted from the global average pooling layer.

2. `LCAuCIDClassifier` Class

- **Purpose:** Extends the `LwPClassifier` to introduce additional functionality for prototype adaptation and regularization.

- **Key Features:**
  - `Initialization` (__init__): Accepts `num_classes`, a regularization parameter (`reg`), and an adaptation parameter (`adt`), which are passed to the parent class.
  - `update` Method: Updates prototypes using input data and pseudo-labels. Further adapts the prototypes by invoking the `lcud` method with the `adt` parameter, balancing stability and adaptation.

3. `lcud` Method

- **Purpose:** Adapts prototypes to minimize shifts between their old and new distributions.
- **Key Features:**
  - Computes a weighted average between `old_prototypes` and `new_prototypes`, controlled by a parameter `alpha`.
  - Ensures smooth transitions in prototype updates, improving stability during adaptation.
- **Output:** Returns a numpy array of adapted prototypes, blending past knowledge (`old_prototypes`) with updates (`new_prototypes`).

4. `Pseudolabels`

- **Definition:** Pseudolabels are labels predicted by the classifier for unlabeled datasets. They serve as approximate labels to enable supervised learning on datasets with missing or unknown ground truth.
- **Usage:** During iterative training, pseudolabels were generated for datasets without true labels. These pseudolabels were then used to update the classifier prototypes.
- **Confidence Thresholding:** A dynamic confidence threshold was applied to ensure that only high-confidence pseudolabels contributed to prototype updates, reducing the impact of noisy predictions.

5. `Targets`

- **Definition:** Targets represent the true labels for the training datasets. When available, they are used directly for supervised training.
- **Handling Missing Targets:** For datasets without labeled targets, pseudolabels were substituted to enable training.
- **Significance:** Targets (or pseudolabels) are integral for the classifier's ability to align prototypes with the correct class distributions.

6. `Feature Extraction`

- **Pipeline:** Features were extracted using a pre-trained MobileNet model, with the top classification layer removed and global average pooling applied.
- **Preprocessing:** Input images were resized to 224×224, normalized to the MobileNet input format, and then processed through the network to obtain 256-dimensional feature vectors.
- **Purpose:** Extracted features capture high-level semantic information, enabling effective prototype-based classification.
- **Batch Processing:** To handle large datasets efficiently, feature extraction was performed in batches of size 32.

7. `Reduced Features`

- **Definition:** Reduced features refer to the 256-dimensional representations output by the MobileNet feature extractor.
- **Storage:** Extracted features were stored for each dataset, enabling reuse during evaluation without recomputation.

- **Significance:** These features provided a compact and meaningful representation of the input data, which was crucial for accurate classification and prototype updates.

Training and Evaluation

- Starting with f_{10} from Task 1.1, models $f_{11}, f_{12}, \ldots, f_{20}$ were iteratively trained on datasets $D_{11}, \ldots, D_{20}$.
- Each model was evaluated on held-out datasets $D_1, \ldots, D_{20}$.

Results

The accuracy matrix for Task 1.2 is shown below:

```
[0.6796 0.8552 0.884  0.9032 0.9268 0.944  0.9504 0.9648 0.9808 0.9888
  0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.6488 0.8384 0.8588 0.876  0.8932 0.91   0.912  0.9284 0.9376 0.9456
  0.9372 0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.6232 0.8028 0.8272 0.8424 0.8536 0.878  0.8708 0.8868 0.9076 0.912
  0.8952 0.8552 0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.626  0.7964 0.8208 0.8404 0.852  0.8704 0.8708 0.8748 0.9044 0.9008
  0.8852 0.8356 0.9528 0.     0.     0.     0.     0.     0.     0.    ]
 [0.6312 0.7972 0.8256 0.8316 0.852  0.8664 0.8692 0.876  0.902  0.9032
  0.8912 0.8612 0.9548 0.9732 0.     0.     0.     0.     0.     0.    ]
 [0.6364 0.8072 0.828  0.8428 0.86   0.8776 0.8776 0.896  0.9116 0.9088
  0.8936 0.8888 0.9472 0.9568 0.9828 0.     0.     0.     0.     0.    ]
 [0.624  0.7916 0.8212 0.8288 0.8448 0.8628 0.8568 0.8772 0.8904 0.8884
  0.8668 0.852  0.9104 0.9524 0.964  0.926  0.     0.     0.     0.    ]
 [0.626  0.7928 0.8236 0.826  0.8396 0.8612 0.8552 0.8744 0.8884 0.8852
  0.8604 0.8848 0.92   0.9392 0.964  0.9444 0.9764 0.     0.     0.    ]
 [0.6204 0.7816 0.8084 0.8076 0.8272 0.8372 0.8404 0.8568 0.8736 0.872
  0.8476 0.8536 0.908  0.9296 0.9528 0.942  0.9632 0.964  0.     0.    ]
 [0.6112 0.7584 0.7764 0.7928 0.8092 0.8176 0.82   0.8392 0.8536 0.8544
  0.8248 0.8032 0.862  0.9096 0.9172 0.8836 0.924  0.9264 0.9216 0.    ]
 [0.6184 0.7724 0.7888 0.7956 0.8136 0.828  0.828  0.8456 0.8636 0.8624
  0.8296 0.8236 0.8736 0.9072 0.9216 0.8972 0.928  0.9348 0.9444 0.968 ]]
```

Figure 2: Accuracy matrix for Task 1.2 showing performance of models $f_{11}, \ldots, f_{20}$ on held-out datasets $D_1, \ldots, D_{20}$.

Conclusion

Our approach effectively handled both similar and varying distributional settings, leveraging MobileNet, PCA, and distillation-based classifiers. Future improvements could explore advanced confidence mechanisms and adaptive dimensionality reduction techniques for further robustness.

PRESENTATION

Final Report: Lifelong Domain Adaptation via Consolidated Internal Distribution (NeurIPS 2021)

This presentation summarizes the problem of lifelong unsupervised domain adaptation, focusing on the proposed method of leveraging a consolidated internal distribution to enable effective model adaptation across domains without requiring labeled data. Key insights, methodologies, and results are discussed concisely in the video. Watch the presentation here
`https://youtu.be/pxMtqf5cGrU`