Sequential Learning on CIFAR-10

Kaneez Fatima
Roll No. 220496

Nikita
Roll No. 220712

Sangam Gupta
Roll No. 220961

Introduction

This mini-project involves exploring the effects of feature distributions and updating methods on model performance in sequential learning tasks using the CIFAR-10 dataset. We are provided with 20 datasets: D1 to D10 share the same input distribution, while D11 to D20 come from different but related distributions. Our goal is to train models sequentially and assess their ability to maintain and improve performance across these datasets.

In Task 1, we train an initial model using the labeled dataset D1 and iteratively update it with predicted labels from the subsequent datasets (D2 to D10), ensuring that performance degradation on earlier datasets is minimized. Task 2 extends this process to datasets D11 to D20, where input distributions vary, requiring adaptations to account for these differences.

Throughout the project, we will evaluate models on held-out datasets to measure their accuracy and report results in a matrix format. The deliverables include well-documented Python notebooks, a detailed report explaining the approach, and a video presentation summarizing a relevant research paper.

Problem 1

Task 1: Sequential Model Training and Evaluation

Objective

The objective of Task 1 is to train sequential machine learning models on datasets D_1 to D_{10}, where D_1 is labeled and the remaining datasets are unlabeled. These datasets share the same input distribution $p(x)$. We aim to iteratively train models $f_1, f_2, \ldots, f_{10}$ such that each subsequent model incorporates knowledge from the previous dataset while maintaining high performance on both the current and prior datasets. This is achieved through a continual learning process that prevents performance degradation on earlier datasets.

Methodology

1. **Feature Extraction:**
   - A pre-trained ResNet-152 model was employed to extract features from the input images. The model's final layer was removed to use its output as high-level feature representations of the input images.
   - Input images were preprocessed using the following transformations:
     - Resizing to 224×224 dimensions.
     - Normalizing using mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$, consistent with the ImageNet dataset.
2. **Prototype-Based Classification:**
   - For the labeled dataset D_1, class prototypes were computed as the mean feature vector for each class. These prototypes serve as class representatives for classification.

- For the unlabeled datasets D_2 to D_{10}, labels were predicted using the nearest prototype classifier, where each input is assigned the label of the closest prototype based on pairwise Euclidean distances.
- After assigning labels, new prototypes were computed iteratively to refine class representations.

3. **Evaluation:**
   - Models $f_1, f_2, \ldots, f_{10}$ were evaluated on held-out datasets $\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_{10}$.
   - Each model f_i was tested on $\hat{D}_i$ as well as all previous held-out datasets $\hat{D}_j$ for $j < i$, ensuring that the performance on earlier datasets did not degrade.
   - Evaluation results were recorded as a matrix, where rows represent models f_i and columns represent the held-out datasets.

Results

- The models were evaluated based on their accuracy on the held-out datasets.
- A continual learning approach was implemented to balance performance between the current and earlier datasets.
- The final model f_{10} achieved high accuracy on $\hat{D}_{10}$ while retaining strong performance on $\hat{D}_1, \ldots, \hat{D}_9$. Prototypes for f_{10} were saved for further use.
- The following accuracy matrix shows the performance of each model f_i on the corresponding held-out datasets:

Table 1: Accuracy Matrix for Task 1

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	0.8988	-	-	-	-	-	-	-	-	-
f_2	0.8828	0.8972	-	-	-	-	-	-	-	-
f_3	0.8772	0.8940	0.8828	-	-	-	-	-	-	-
f_4	0.8732	0.8928	0.8796	0.8824	-	-	-	-	-	-
f_5	0.8736	0.8928	0.8776	0.8796	0.8872	-	-	-	-	-
f_6	0.8720	0.8912	0.8756	0.8776	0.8832	0.8812	-	-	-	-
f_7	0.8744	0.8900	0.8768	0.8756	0.8828	0.8812	0.8784	-	-	-
f_8	0.8648	0.8824	0.8716	0.8720	0.8744	0.8772	0.8668	0.8736	-	-
f_9	0.8600	0.8808	0.8672	0.8704	0.8736	0.8724	0.8668	0.8740	0.8676	-
f_{10}	0.8632	0.8852	0.8692	0.8700	0.8768	0.8764	0.8684	0.8756	0.8672	0.8684

Insights

The iterative prototype-based method provided an efficient way to incorporate knowledge from sequential datasets. By leveraging a pre-trained ResNet-152 for feature extraction, the computational complexity was reduced while maintaining robust classification performance. This method highlights the importance of maintaining class representations across sequential tasks to mitigate catastrophic forgetting.

Task 2: Sequential Model Training on Diverse Distributions

Objective

The goal of Task 2 is to train sequential models on datasets D_{11} to D_{20}, which originate from distributions that differ slightly from the shared input distribution of D_1 to D_{10}. These datasets introduce domain shifts, requiring adaptive methods to ensure robust performance while preventing significant degradation on previous datasets. Starting with the prototypes from f_{10} (Task 1), we aim to iteratively fine-tune models $f_{11}, f_{12}, \ldots, f_{20}$ to address these challenges.

Methodology

1. **Feature Extraction and Preprocessing:**
   - A pre-trained ResNet-152 model was used for feature extraction, with fine-tuning enabled on its final layers to adapt to the domain shift in D_{11} to D_{20}.
   - Input images were preprocessed with transformations similar to Task 1, including resizing, normalization, and conversion to tensors, ensuring consistency across tasks.
   - Features and prototypes were normalized to prevent numerical instability during computation.
2. **Fine-Tuning the Feature Extractor:**
   - The feature extractor was fine-tuned on each dataset using a labeled subset, optimizing the parameters of its final layers for better adaptation to new distributions.
   - The Adam optimizer and cross-entropy loss were used during training, with five epochs per dataset to balance performance and computational cost.
3. **Prototype Adaptation:**
   - For labeled datasets, class prototypes were updated using weighted averages of current and new prototypes. This blending ensured stability across domains, with a weight factor ($\alpha = 0.5$) balancing previous knowledge and current adaptations.
   - For unlabeled datasets, predicted labels were assigned using the nearest prototype classifier, and prototypes were computed based on these predictions.
   - Prototypes were normalized after every update to maintain consistency across feature representations.
4. **Evaluation:**
   - Models $f_{11}, f_{12}, \ldots, f_{20}$ were evaluated on held-out datasets $\hat{D}_{11}, \hat{D}_{12}, \ldots, \hat{D}_{20}$ and all prior datasets $\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_{20}$.
   - Evaluation results were recorded in a matrix, with rows representing models f_i and columns representing the datasets.

Results

- Fine-tuning the feature extractor improved adaptability to the domain shift in D_{11} to D_{20}, achieving high accuracy on corresponding held-out datasets.
- Weighted prototype updates effectively balanced new domain knowledge with prior representations, minimizing performance degradation on earlier datasets.
- The final model f_{20} maintained competitive performance across all held-out datasets, highlighting the effectiveness of sequential learning and domain adaptation strategies.
- The following accuracy matrix shows the performance of each model f_{i+10} on the corresponding held-out datasets:

Table 2: Accuracy Matrix for Task 2

Model	$\hat{D}_1$	$\hat{D}_2$	$\hat{D}_3$	$\hat{D}_4$	$\hat{D}_5$	$\hat{D}_6$	$\hat{D}_7$	$\hat{D}_8$	$\hat{D}_9$	$\hat{D}_{10}$	$\hat{D}_{11}$	$\hat{D}_{12}$	$\hat{D}_{13}$	$\hat{D}_{14}$	$\hat{D}_{15}$	$\hat{D}_{16}$	$\hat{D}_{17}$	$\hat{D}_{18}$	$\hat{D}_{19}$	$\hat{D}_{20}$
f_{11}	0.4420	0.4452	0.4400	0.4432	0.4376	0.4396	0.4352	0.4472	0.4324	0.4468	0.3680	-	-	-	-	-	-	-	-	-
f_{12}	0.5612	0.5384	0.5568	0.5684	0.5472	0.5584	0.5548	0.5588	0.5292	0.5596	0.4360	0.3508	-	-	-	-	-	-	-	-
f_{13}	0.7324	0.7364	0.7408	0.7488	0.7328	0.7504	0.7432	0.7336	0.7272	0.7460	0.5956	0.4644	0.6488	-	-	-	-	-	-	-
f_{14}	0.8000	0.8080	0.7984	0.8128	0.7972	0.8148	0.8024	0.8024	0.7952	0.8056	0.6688	0.5060	0.7108	0.7840	-	-	-	-	-	-
f_{15}	0.8180	0.8304	0.8112	0.8324	0.8148	0.8304	0.8212	0.8200	0.8132	0.8220	0.6864	0.5164	0.7284	0.7952	0.8240	-	-	-	-	-
f_{16}	0.8224	0.8332	0.8180	0.8380	0.8196	0.8312	0.8252	0.8244	0.8140	0.8264	0.6892	0.5052	0.7340	0.8008	0.8260	0.7120	-	-	-	-
f_{17}	0.8288	0.8428	0.8256	0.8396	0.8304	0.8368	0.8300	0.8272	0.8200	0.8256	0.6908	0.4952	0.7284	0.8056	0.8344	0.7112	0.7796	-	-	-
f_{18}	0.8300	0.8404	0.8272	0.8404	0.8324	0.8360	0.8336	0.8332	0.8220	0.8284	0.6864	0.5020	0.7324	0.8112	0.8296	0.7132	0.7808	0.6984	-	-
f_{19}	0.8172	0.8352	0.8192	0.8328	0.8280	0.8276	0.8244	0.8268	0.8172	0.8216	0.6676	0.5056	0.7268	0.7992	0.8224	0.6956	0.7628	0.6964	0.5844	-
f_{20}	0.8264	0.8392	0.8292	0.8372	0.8348	0.8364	0.8332	0.8368	0.8252	0.8296	0.6824	0.5048	0.7348	0.8056	0.8316	0.7096	0.7736	0.7044	0.5932	0.8088

Insights

Task 2 demonstrates the importance of fine-tuning and adaptive prototype-based methods for handling domain shifts in sequential learning. By combining feature normalization, fine-tuning, and weighted prototype updates, the models effectively adapted to new distributions while preserving past performance. This approach highlights the challenges and solutions in continual learning for diverse input distributions.

Problem 2: Deja vu: Continual model generalization for unseen domains (ICLR 2023)

You tube Link:

https://youtu.be/UZDoqQXh3Yg?si=LIDbPwgmjm6vSLHY

References

Snell, J., Swersky, K., Zemel, R. S. (2017). Prototypical Networks for Few-Shot Learning. Advances in Neural Information Processing Systems :- https://arxiv.org/abs/1703.05175

Wang, W., Yan, Y., Sebe, N. (2020). Lifelong and Continual Learning: A Survey :- https://arxiv.org/abs/2007.15603

He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) :- https://arxiv.org/abs/1512.03385