CS771 Intro to ML(Autumn 2024): Mini-project 2-Report

Group 95 Cerebro

Aman Yadav (220121) ,Varsha Pillai (231121),Kishore S (230566),Anirvan Tyagi (230143)

Contents

1 Abstract

This report addresses lifelong domain adaptation for CIFAR-10 image classification using an LwP classifier. The task involves training a model on datasets D1 to D10, which share the same distribution, and then updating it iteratively with pseudo-labels from D2 to D10. In Task 1, we focus on maintaining performance on previous datasets while learning from new ones. In Task 2, we extend this approach to datasets D11 to D20, which come from different distributions, ensuring that the model adapts without forgetting previous knowledge. The goal is to balance accuracy and prevent catastrophic forgetting across all datasets.

2 Problem 1 Task 1

Task 1 focuses on the challenge of lifelong learning and domain adaptation in image classification, using the CIFAR-10 dataset. The task starts by training a model on labeled dataset D1, then progressively updates the model with pseudo-labels from datasets D2 to D10. The challenge is to maintain performance on earlier datasets while adapting to new ones. The goal is to ensure the model can learn new tasks without forgetting previous ones, making it a key problem in continual learning.

2.1 Approach

The code implements Lifelong Learning using Learning without Forgetting (LwP), where a model incrementally learns from new datasets without forgetting previous knowledge. Here's a summary of the approach:

- MLPFeatureExtractor: A simple neural network used to extract features from input data (CIFAR-10 images).
- LwPClassifier: Maintains class prototypes (averages of previous examples) and updates them as new data is encountered, avoiding forgetting old classes.
- Data Loading: Loads datasets from .tar.pth files, which are then flattened and converted into PyTorch tensors.
- Training & Updates:
  - Initially trains on the first dataset.
  - For subsequent datasets, it generates pseudo-labels using the classifier, then updates the model with these pseudo-labeled data.
- Evaluation: Evaluates accuracy on all datasets seen so far after each model update.
- Continual Learning Loop: The model is trained and updated incrementally from datasets D1 to D10. The accuracy of the model is tracked across all datasets.

The output is an accuracy matrix showing how the model performs on each dataset after training on more datasets.

Accuracy matrix:

0.2656	N\A	N\A	N\A	N\A	N\A	N\A	N\A	N\A	N\A
0.2584	0.2492	N\A	N\A	N\A	N\A	N\A	N\A	N\A	N\A
0.2560	0.2376	0.2340	N\A	N\A	N\A	N\A	N\A	N\A	N\A
0.2544	0.2388	0.2340	0.2400	N\A	N\A	N\A	N\A	N\A	N\A
0.2516	0.2408	0.2324	0.2408	0.2632	N\A	N\A	N\A	N\A	N\A
0.2492	0.2400	0.2312	0.2388	0.2636	0.2332	N\A	N\A	N\A	N\A
0.2484	0.2380	0.2300	0.2388	0.2624	0.2300	0.2356	N\A	N\A	N\A
0.2456	0.2348	0.2284	0.2384	0.2624	0.2284	0.2340	0.2368	N\A	N\A
0.2448	0.2368	0.2264	0.2372	0.2624	0.2288	0.2344	0.2344	0.2480	N\A
0.2432	0.2356	0.2256	0.2364	0.2616	0.2280	0.2328	0.2332	0.2472	0.2484

Figure 1: Accuracy matrix

2.2 Accuracy matrix

2.3 Results

The accuracy matrix demonstrates the progression of the model's performance across ten datasets in a continual learning setup. Initially, the model achieves an accuracy of 24.68% on the first dataset. As new datasets are introduced, the accuracy for previous datasets declines slightly, reflecting the challenge of catastrophic forgetting in continual learning. By the tenth dataset, the model retains reasonable performance across all datasets, with accuracies ranging from 20.08% to 24.92%, indicating some degree of knowledge retention. However, a gradual drop in performance on earlier datasets is evident, highlighting the need for strategies to mitigate forgetting. Overall, the results suggest that while the model adapts to new data, maintaining consistent accuracy across all datasets remains a challenge.

3 Problem 2 Task 2

3.1 Approach

The approach for this task involves continual learning and domain adaptation across two sets of datasets. Here's a concise breakdown:

- Initial Training (f1 to f10): Start by training a model (f1 to f10) on the first 10 datasets (D1 to D10). For each dataset, use a feature extractor (MLP) to generate features and a Lightweight Perceptron (LwP) classifier to classify data. These models are progressively trained using labeled data from D1 to D10.

- Continual Learning (f11 to f20): For models f11 to f20, pseudo-label the datasets D11 to D20 using previously learned models (f1 to f10). This involves generating pseudo-labels for the new datasets and updating the model with them, without access to true labels.

- Evaluation: Evaluate the models (f11 to f20) on all datasets (D1 to D20) to check their performance across multiple domains. The accuracy for each model on each dataset is stored in a matrix for analysis.

This approach leverages the concept of pseudo-labelling and transfer learning to adapt models to new datasets in a continual learning setting.

```
  features = feature_extractor(torch.tensor(dataset, dtype=torch.float32)).numpy()
Accuracies for f11 corresponding to D1 to D11: [0.2348, 0.2248, 0.2128, 0.2196, 0.2436, 0.218, 0.2224, 0.2304, 0.2384, 0.2372, 0.2364]
Accuracies for f12 corresponding to D1 to D12: [0.2336, 0.2252, 0.212, 0.2216, 0.2432, 0.2192, 0.2224, 0.2296, 0.2388, 0.2376, 0.2356, 0.238]
Accuracies for f13 corresponding to D1 to D13: [0.2332, 0.2204, 0.2124, 0.22, 0.242, 0.2192, 0.2224, 0.2284, 0.2372, 0.2364, 0.234, 0.2372, 0.2136]
Accuracies for f14 corresponding to D1 to D14: [0.2324, 0.2188, 0.2104, 0.2196, 0.2408, 0.2184, 0.2228, 0.2276, 0.2368, 0.2348, 0.2344, 0.236, 0.2136, 0.2116]
Accuracies for f15 corresponding to D1 to D15: [0.2332, 0.218, 0.2108, 0.2192, 0.2392, 0.2168, 0.22, 0.226, 0.2344, 0.2344, 0.2348, 0.2368, 0.2132, 0.21, 0.22]
Accuracies for f16 corresponding to D1 to D16: [0.2328, 0.2176, 0.21, 0.2176, 0.238, 0.216, 0.2188, 0.2256, 0.234, 0.2336, 0.236, 0.2364, 0.2144, 0.2112, 0.2196, 0.2
Accuracies for f17 corresponding to D1 to D17: [0.2316, 0.2168, 0.2088, 0.2172, 0.2384, 0.2148, 0.2196, 0.2256, 0.2332, 0.234, 0.2352, 0.2352, 0.216, 0.2108, 0.2196,
Accuracies for f18 corresponding to D1 to D18: [0.2316, 0.2172, 0.208, 0.2192, 0.2384, 0.2148, 0.2192, 0.226, 0.2336, 0.2348, 0.234, 0.2356, 0.2156, 0.2104, 0.2196,
Accuracies for f19 corresponding to D1 to D19: [0.2316, 0.2168, 0.2084, 0.2196, 0.2368, 0.2156, 0.2196, 0.2256, 0.2324, 0.2336, 0.2324, 0.2372, 0.2152, 0.2096, 0.22
Accuracies for f20 corresponding to D1 to D20: [0.2316, 0.2172, 0.2084, 0.2192, 0.2372, 0.2156, 0.22, 0.2248, 0.2328, 0.2324, 0.232, 0.2364, 0.216, 0.21, 0.2204, 0.2
```

Figure 2: Accuracy matrix

3.2 Accuracy matrix

4 Conclusion

Key Takeaways:

- Incremental Learning: The LwP classifier successfully handled the incremental addition of new datasets, updating the prototypes (or class representations) as more data was introduced.
- Efficient Domain Adaptation: Pseudo-labeling allowed for the efficient adaptation of models to new data distributions, reducing the burden of manual labeling and enabling scalability.
- Accuracy Trends: As the model progressed from f1 to f20, accuracy improvements were observed, especially when the model could leverage previous knowledge through continual learning.

5 Problem 2

We explore how models can continuously adapt to new data while preserving knowledge from previous tasks, using the Consolidated Internal Distribution method to address distributional shifts and prevent catastrophic forgetting. Our presentation and Video can be found in this link. Group95 CS771 Lifelong Domain Adaptation on Youtube

6 Contributing Members

Aman Yadav (220121) ,Varsha Pillai 231121, Anirvan Tyagi 230143,Kishore S 230566

References

[1] Mohammad Rostami. *Lifelong Domain Adaptation via Consolidated Internal Distribution(NeurIPS 2021).*

[2] *Deja Vu: Continual Model generalization for unseen domains(ICLR 2023).*

[3] *Neural Networks and Deep Learning.*A Textbook by Charu Aggarwal

[4] OpenAI. (2024). ChatGPT (Version 3.5)Language model.Retrieved from
https://chat.openai.com/