Binary Classification - Mini Project II

Adarsh Anurag
Roll No. 220053

Rohit Verma
Roll No. 220917

Sambuddha Chakrabarti
Roll No. 220947

Sharah PS
Roll No. 221001

Tanmay Soni
Roll No. 221130

1 Introduction

The increasing complexity of real-world datasets and the need for adaptive models capable of handling evolving distributions present significant challenges in machine learning. This project aims to address these challenges using the CIFAR-10 image classification dataset, divided into subsets with distinct distribution characteristics. The goal is to iteratively train and adapt models across a sequence of datasets while ensuring their performance remains robust across previously encountered data.

- **Task 1** focuses on datasets with similar input distributions, emphasizing a progressive learning approach to train models. Each model is trained using the predicted labels from its predecessor, and the performance is evaluated on a corresponding held-out dataset. A critical objective is to ensure minimal performance degradation on earlier datasets as the models evolve.
- **Task 2** introduces a more complex scenario, where datasets are derived from varying distributions. This requires adapting the training strategy to account for distribution shifts while maintaining model performance across all held-out datasets.

2 Task 1

2.1 Model and Approach

For this task, we used the pre-trained Resnet50 framework to preprocess the inputs, following which we used a prototypical network to find the most suitable prototype for representing this adjusted feature space. ResNet50 is a deep convolutional neural network with 50 layers that employs a residual learning framework to ease the training of very deep networks. ResNet50's residual architecture is excellent at learning hierarchical feature representations, making it suitable for extracting features even from small-scale datasets like CIFAR. ResNet50 is trained on similar image data, (ImageNet), and hence it is suitable for extracting deeper features for classification and rejecting noise from the inputs.

The prototypical network uses an input layer of 2048, the new transformed feature space, and is then bringing the dimensions down to 128, which forms the new embedding. It is trained using the labels and is motivated to reduce intra-class variance while increasing inter-class distance. We were able to obtain an accuracy of 60% on average for the datasets. For the prototype computation, we have used Euclidean distance for classifying the new unseen datasets. No data from the earlier datasets are used for updation of model parameters. Only the model is indicative of the previous training done, while any changes made to the parameters reflect the new dataset. .

Table 1: Accuracy Matrix for Models and Held-out Datasets

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	0.6224									
f_2	0.6308	0.6248								
f_3	0.6304	0.6276	0.6096							
f_4	0.6276	0.6160	0.6056	0.6244						
f_5	0.6200	0.6196	0.6040	0.6212	0.6272					
f_6	0.6144	0.6252	0.5960	0.6224	0.6204	0.6140				
f_7	0.6080	0.6236	0.6024	0.6176	0.6196	0.6080	0.6072			
f_8	0.6076	0.6240	0.5948	0.6144	0.6208	0.6052	0.6028	0.5956		
f_9	0.6076	0.6168	0.5940	0.6084	0.6224	0.6084	0.6028	0.5976	0.5956	
f_{10}	0.6036	0.6152	0.5956	0.5984	0.6164	0.6008	0.5980	0.5952	0.5904	0.6072

3 Task 2

3.1 Approach

To train the prototypical network on the 11th dataset while leveraging domain adaptation from the previously trained model, we can use a domain adaptation loss (e.g., Maximum Mean Discrepancy, or MMD) to align embeddings from the 11th dataset with those from the previously trained network. Additionally, we include a prototype consistency loss to ensure the updated prototypes for the first 10 datasets do not deviate significantly from their previous values. However, we observed lesser rates of accuracy when it was tested on the evaluation dataset. Finally, we found the best rates on the entire test evaluation to be on the prototypical network trained on Resnet features

Table 2: Accuracy Matrix for Task 2

Model	$\hat{D}_1$	$\hat{D}_2$	$\hat{D}_3$	$\hat{D}_4$	$\hat{D}_5$	$\hat{D}_6$	$\hat{D}_7$	$\hat{D}_8$	$\hat{D}_9$	$\hat{D}_{10}$
f_{11}	0.5840	0.5868	0.5832	0.5756	0.5960	0.5836	0.5740	0.5796	0.5744	0.5668
f_{12}	0.5712	0.5724	0.5656	0.5612	0.5816	0.5652	0.5564	0.5724	0.5560	0.5552
f_{13}	0.5672	0.5688	0.5616	0.5532	0.5748	0.5660	0.5524	0.5688	0.5536	0.5552
f_{14}	0.5668	0.5668	0.5604	0.5472	0.5748	0.5620	0.5492	0.5632	0.5464	0.5468
f_{15}	0.5640	0.5604	0.5588	0.5532	0.5660	0.5648	0.5456	0.5612	0.5488	0.5520
f_{16}	0.5512	0.5580	0.5388	0.5416	0.5552	0.5492	0.5372	0.5524	0.5408	0.5448
f_{17}	0.5444	0.5576	0.5500	0.5364	0.5556	0.5428	0.5288	0.5460	0.5416	0.5372
f_{18}	0.5356	0.5476	0.5400	0.5220	0.5404	0.5348	0.5236	0.5340	0.5292	0.5320
f_{19}	0.5356	0.5424	0.5376	0.5224	0.5372	0.5416	0.5208	0.5304	0.5232	0.5264
f_{20}	0.5300	0.5464	0.5436	0.5200	0.5356	0.5344	0.5232	0.5284	0.5280	0.5256

Table 3: Accuracy Matrix for Task 2

Model	$\hat{D}_{11}$	$\hat{D}_{12}$	$\hat{D}_{13}$	$\hat{D}_{14}$	$\hat{D}_{15}$	$\hat{D}_{16}$	$\hat{D}_{17}$	$\hat{D}_{18}$	$\hat{D}_{19}$	$\hat{D}_{20}$
f_{11}	0.5228	-	-	-	-	-	-	-	-	-
f_{12}	0.5020	0.3668	-	-	-	-	-	-	-	-
f_{13}	0.4984	0.3644	0.4780	-	-	-	-	-	-	-
f_{14}	0.4980	0.3584	0.4716	0.5012	-	-	-	-	-	-
f_{15}	0.4944	0.3592	0.4688	0.4868	0.5688	-	-	-	-	-
f_{16}	0.4824	0.3716	0.4736	0.4820	0.5596	0.4504	-	-	-	-
f_{17}	0.4948	0.3704	0.4684	0.4828	0.5504	0.4332	0.4440	-	-	-
f_{18}	0.4740	0.3728	0.4684	0.4732	0.5424	0.4400	0.4404	0.4420	-	-
f_{19}	0.4620	0.3724	0.4664	0.4700	0.5384	0.4264	0.4396	0.4312	0.4736	-
f_{20}	0.4652	0.3708	0.4552	0.4700	0.5412	0.4188	0.4380	0.4300	0.4656	0.4788

4 Problem 2

The paper ”Lifelong Domain Adaptation via Consolidated Internal Distribution” presents a way for machine learning models to adapt to new tasks without forgetting what they've already learned. It tackles two main challenges: changes in data patterns (called domain shift) and the loss of past knowledge (catastrophic forgetting). The method, called LDAuCID, keeps track of key patterns from earlier tasks and aligns new data to these patterns. It also reuses important examples from past tasks to help the model remember. This approach works well on standard datasets, allowing the model to learn new tasks while keeping its old knowledge intact, making it useful for real-world situations where data changes over time.

Link to the video

5 Acknowledgments

We sincerely thank Professor Piyush Rai for their invaluable guidance throughout this project. We are also grateful to the open-source Python community for providing the tools essential to our work, and to the machine learning researchers whose work has significantly inspired and contributed to this project.

References

[1] Rostami, M. (2021). ”Lifelong Domain Adaptation via Consolidated Internal Distribution.” *Advances in Neural Information Processing Systems*, 34, 11172-11183. DOI: 5caf41d62364d5b41a893adc1a9dd5d4-Paper.pdf

[2] 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)*, 770-778. DOI: 10.1109/CVPR.2016.90

[3] Bishop, C. M. (2006). ”Pattern Recognition and Machine Learning.” *Springer*, 1st Edition. DOI: 10.1007/978-0-387-45528-0

[4] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). ”Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research*, 12, 2825-2830. Available at: http://jmlr.org/papers/v12/pedregosa11a.html

[5] Van Rossum, G., & Drake, F.L. (2009). *Python 3 Reference Manual*. CreateSpace. Available at: https://docs.python.org/3/