CS771 Mini Project 2 - Group 67 Transformers

Aayush Singh
220024

Aditya Gautam
220064

Akash Verma
220097

Ritik Shah
220894

Samarth Agarwal
220944

Problem 1

1 Introduction and Overall Approach

The task involves using 20 provided training datasets, each of which is a subset of the CIFAR-10 image classification dataset. The true labels are only available for the first dataset. For the subsequent datasets, we need to generate pseudo-labels to update the model. Given that we are constrained to using a Learning with Prototypes (LwP)-based model, the key challenge lies in how and when we update the prototypes to ensure an overall satisfactory accuracy and minimal forgetting.

The following report gives an overview of the development of our final model, while concentrating on the elements of the final model and provides the final results. We extracted features using the ResNet50 Feature Extractor. We then used these extracted features to perform LwP.

Task 1

2 Task 1 approach

2.1 Feature Extraction and the Model

In task one, the input probability distribution of Dataset 1 to 10 are similar, and we have labels available for only Dataset 1. We used a ResNet50Extractor to extract features from the Datasets. The model we used was an LwP model which used these extracted features to make predictions.

2.2 Continual training on Dataset 1 to Dataset 10

We first did initial training on Dataset 1. Since the labels were available, this was supervised learning. We trained the first model on Dataset 1 (f1) without using distillation loss.

For the subsequent models, we used the `self_train_update` function present in the submitted notebook to train the models using its unlabelled data and the model from the previous dataset. We used distillation loss here which greatly improved the accuracy.

2.3 Losses Used and Teacher-Student Analogy

Here the new model can be seen as a student, while the previous model that is being used to generate the new model in addition to the corresponding unlabelled datasets is called the student model. The student is trained on the pseudo-labeled data generated using the teacher-model. The total loss includes:

- Cross-Entropy Loss: Matches the student's predictions to the pseudo-labels.
- Prototype Loss: Encourages alignment of the student's features with the prototypes.
- Distillation Loss: Aligns the student's outputs with the teacher's soft labels and enables knowledge distillation.

2.4 Further innovation to take the weighted average of the prototypes

While moving from dataset n to dataset n+1, once a model is trained on dataset n+1, we decided to take a weighted average with the prototypes of the nth model. Essentially, this is like creating new_prototype_final = $\frac{n-1}{n}$ * old_prototype + $\frac{1}{n}$ * new_prototype_initial. This helped in better retention and prevention of catastrophic forgetting even further ahead of knowledge distillation.

2.5 Summary of Components of the Final Approach for Task 1

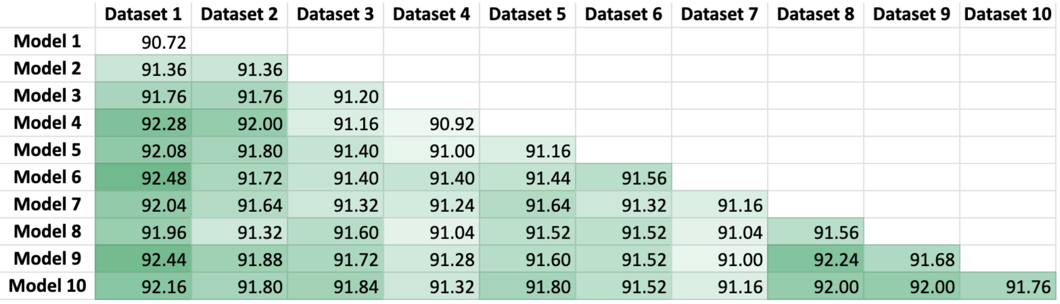
The final approach included the following:

ResNet50 extractor + LwP (Euclidean Distance based classifier layer) along with distillation, prototype and CE loss along with an additional action of taking the weighted average of the previous model's prototypes.

2.6 Results and Conclusion for Task 1

Because of the identical input probability distribution of the datasets, good enough results were obtained from this simple implementation. Trying further advanced techniques from RaTP did not yield much improvement, so we selected this simple code as given in the attached notebook as our final model. In conclusion, we obtain an acceptable accuracy matrix using knowledge distillation and weighted averaging of prototypes.

Figure 1: Result for Task 1



	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8	Dataset 9	Dataset 10
Model 1	90.72									
Model 2	91.36	91.36								
Model 3	91.76	91.76	91.20							
Model 4	92.28	92.00	91.16	90.92						
Model 5	92.08	91.80	91.40	91.00	91.16					
Model 6	92.48	91.72	91.40	91.40	91.44	91.56				
Model 7	92.04	91.64	91.32	91.24	91.64	91.32	91.16			
Model 8	91.96	91.32	91.60	91.04	91.52	91.52	91.04	91.56		
Model 9	92.44	91.88	91.72	91.28	91.60	91.52	91.00	92.24	91.68	
Model 10	92.16	91.80	91.84	91.32	91.80	91.52	91.16	92.00	92.00	91.76

Task 2

3 Task 2 approach

3.1 Feature Extraction and the Model

In task two, we call the 10 datasets given as Dataset 11 to 20. The input probability distribution of Dataset 11 to 20 is different from each other. We have again used a ResNet50Extractor to extract features from the Datasets. The model we used was an LwP model which used these extracted features to make predictions. But we have made some additional changes, especially the usage of T2PL, to improve the performance.

Using the same approach as Task-1 worsens the accuracy, as all the input distributions are different, and it will result in far from ideal prototypes.

3.2 Top2 Pseudo Labeling(`T2PL`)

Essentially, we have applied the `T2PL` method explained in detail in the paper Deja Vu: Continual Model Generalization For Unseen Domains bu Liu et.al. We use only those examples in which probability of the example belonging to the top 50% of samples per class, this helps by removing uncertain examples, and thereby improving accuracy. We have also used a KNN classifier for further refinement of pseudo-labels. Complete information about the `T2PL` implementation is present in the `self_train_update_with_t2pl` function in the attached notebook for task 2.

3.3 Random Mixture and further effect of Weighted Prototypes

We tried a Random Mixture Approach, in which we inject noise to improve robustness of the model using adaptive instance normalization (AdaIN) data augmentation. The expectation was that injecting noise in earlier model would move the prototypes somewhat closer to the ideal prototypes for later datasets, this method was not much effective which was also suggested by the paper *Deja Vu: Liu et.al*. Hence we did not use Random Mixture.

We also performed the additional action of taking the weighted average of the previous model's prototypes, in creating the current model's prototypes. Essentially, this is like creating $\texttt{new\_prototype\_final} = \frac{n-1}{n} \star \texttt{old\_prototype} + \frac{1}{n} \star \texttt{new\_prototype\_initial}$. When we implemented our model for task 2 with weighted averaging of prototypes, considerable improvements were seen. For example, in the 20th model, on the 19th dataset, an improvement of approx 10% was obtained after using the weighted average prototypes. Hence we performed the weighted averaging in our final result.

3.4 Summary of Components of the Final Approach for Task 2

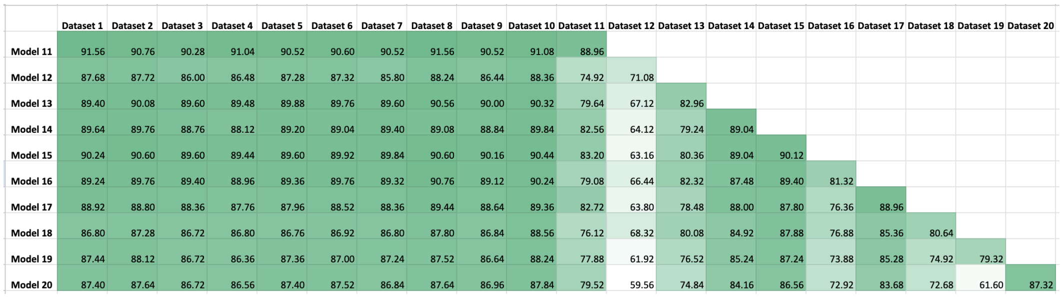
The final approach included the following:

ResNet50 extractor + LwP (Euclidean distance-based classifier layer) along with distillation, prototype and CE loss and Top2 Pseudo Labelling, along with an additional action of taking the weighted average of the previous model's prototypes.

3.5 Results

In conclusion, we obtain an acceptable accuracy matrix using knowledge distillation, `T2PL` and weighted averaging of prototypes.

Figure 2: Result for Task 2



	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8	Dataset 9	Dataset 10	Dataset 11	Dataset 12	Dataset 13	Dataset 14	Dataset 15	Dataset 16	Dataset 17	Dataset 18	Dataset 19	Dataset 20
Model 11	91.56	90.76	90.28	91.04	90.52	90.60	90.52	91.56	90.52	91.08	88.96									
Model 12	87.68	87.72	86.00	86.48	87.28	87.32	85.80	88.24	86.44	88.36	74.92	71.08								
Model 13	89.40	90.08	89.60	89.48	89.88	89.76	89.60	90.56	90.00	90.32	79.64	67.12	82.96							
Model 14	89.64	89.76	88.76	88.12	89.20	89.04	89.40	89.08	88.84	89.84	82.56	64.12	79.24	89.04						
Model 15	90.24	90.60	89.60	89.44	89.60	89.92	89.84	90.60	90.16	90.44	83.20	63.16	80.36	89.04	90.12					
Model 16	89.24	89.76	89.40	88.96	89.36	89.76	89.32	90.76	89.12	90.24	79.08	66.44	82.32	87.48	89.40	81.32				
Model 17	88.92	88.80	88.36	87.76	87.96	88.52	88.36	89.44	88.64	89.36	82.72	63.80	78.48	88.00	87.80	76.36	88.96			
Model 18	86.80	87.28	86.72	86.80	86.76	86.92	86.80	87.80	86.84	88.56	76.12	68.32	80.08	84.92	87.88	76.88	85.36	80.64		
Model 19	87.44	88.12	86.72	86.36	87.36	87.00	87.24	87.52	86.64	88.24	77.88	61.92	76.52	85.24	87.24	73.88	85.28	74.92	79.32	
Model 20	87.40	87.64	86.72	86.56	87.40	87.52	86.84	87.64	86.96	87.84	79.52	59.56	74.84	84.16	86.56	72.92	83.68	72.68	61.60	87.32

4 Further Work

Throughout our research, we realised that ViT was another feature extractor that could have given decent on the given task. Other models, wherein the nodes have a large capacity (to remember the past and learn the coming data) can be explored. Additionally, other modelling and distance techniques could also be explored.

Problem 2

The link to our video on the paper ”Deja Vu: Continual Model Generalization For Unseen Domains” by Liu et.al. is this: `https://youtu.be/p72HD_CpqiQ`

Acknowledgements

We thank Prof. Piyush Rai for his guidance and explaining the concepts in a well defined manner and the TAs of the course for their support in execution of the course.

References

- The Scikit Learn Library and their models and other tools.
- Pytorch, Resnet, and other ML python libraries.
- Prof. Piyush Rai’s lectures
- DEJA VU: CONTINUAL MODEL GENERALIZATION FOR UNSEEN DOMAINS
- Lifelong Domain Adaptation via Consolidated Internal Distribution