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

ML Group 24:

1. Nishvaan Sai H - 200648
2. Krishna Kumar Bais - 241110038
3. Sevak Baliram - 241110065
4. Rohan - 241110057
5. Amit Kumar Meena - 220126

1 Problem 1

1.1 Introduction

This project explores the challenges and methodologies of continual learning in the context of image classification using the CIFAR-10 dataset. The problem is structured into two tasks: incremental model updates and domain adaptation. In Task 1, we sequentially train and update models on datasets originating from a single input distribution, ensuring minimal degradation in performance on previous datasets. Task 2 extends this idea to datasets with varying input distributions, emphasising the adaptation of models to new domains while retaining previously acquired knowledge. The project leverages a Learning with Prototypes (LwP) classifier and investigates strategies for generative replay and prototype regularisation to achieve robust continual learning performance. The findings contribute to understanding continual learning's potential to handle non-stationary data distributions effectively.

1.1.1 Objective

The primary objectives of this project are:

1. To train and incrementally update models on sequential datasets while minimising catastrophic forgetting and maintaining performance on prior datasets.
2. To adapt the methodology to datasets with varying input distributions, ensuring effective domain adaptation and model generalisation.
3. To evaluate and analyse model performance through a structured accuracy matrix across multiple tasks and datasets, highlighting continual learning's efficacy and limitations.
4. To demonstrate the application of generative classifiers and prototype-based learning in a practical setting, drawing insights from state-of-the-art research in lifelong learning and unsupervised domain adaptation.

1.2 Task 1

This task involves continually training datasets 1 to 10. There is no need for domain adaptation because all the datasets have the same or similar distribution.

1.2.1 Feature Extraction Approach

In this task, feature extraction was performed using a pre-trained InceptionV3 model from the TensorFlow library. The model, configured without the top classification layer (`include_top=False`), processes input images resized to 299×299 pixels. The extracted features are pooled using an average pooling operation with an 8×8 window, resulting in a compact and meaningful feature representation for each image.

The datasets, consisting of raw 32×32 RGB images, were handled using NumPy for data manipulation and TensorFlow's resizing utilities to scale them to the required dimensions. A custom data generator was implemented to efficiently preprocess images in batches, ensuring the smooth handling of large datasets. The generator was resized and normalised (using InceptionV3's specific preprocessing function), and batches of data were prepared for feature extraction.

Using the extracted features, each dataset was processed independently. The features were flattened into one-dimensional vectors, enabling their use in subsequent training tasks. Python's global namespace was utilised to dynamically manage datasets for iterative processing, allowing seamless integration of extracted features from multiple datasets into the workflow.

Essential libraries employed include:

- **TensorFlow**: This is used to utilise the pre-trained InceptionV3 model and image preprocessing.
- **NumPy**: For efficient array manipulation.
- **Torch**: This is for loading datasets in `.pth` format.
- **Math**: This is used to calculate processing steps based on batch sizes.

This systematic approach ensured consistent and efficient extraction of robust features from the CIFAR-10 subsets, forming the foundation for model training and evaluation.

1.2.2 Continual Learning Framework

The model employed for this task is a **Generative Classifier with continuous adaptation**, designed to handle sequential datasets while mitigating catastrophic forgetting. The classifier uses Gaussian class conditions with parameters (means and covariances) that are updated incrementally using new data and prototype regularisation. Generative replay ensures the retention of prior knowledge by synthesising samples from learned distributions of previous tasks.

1.2.2.1 Workflow for Continual Adaptation

1. **Initialisation**: The first model is trained on the D1 dataset after standardising features using `StandardScaler` and reducing dimensionality with PCA. The Gaussian parameters (means and covariances) are calculated for each class.
2. **Sequential Updates**: For each subsequent dataset:
    - The features are standardised and transformed with the previously fitted PCA.
    - Synthetic replay data is generated using the Gaussian parameters of the prior model.
    - The replay data is combined with the current dataset to form a composite training set.
    - A new classifier is trained using this composite dataset. Regularisation ensures smooth transitions by blending the parameters of the previous and current datasets.
3. **Prediction and Evaluation**: Each model predicts labels for the respective datasets, and performance is evaluated, ensuring stability across tasks.

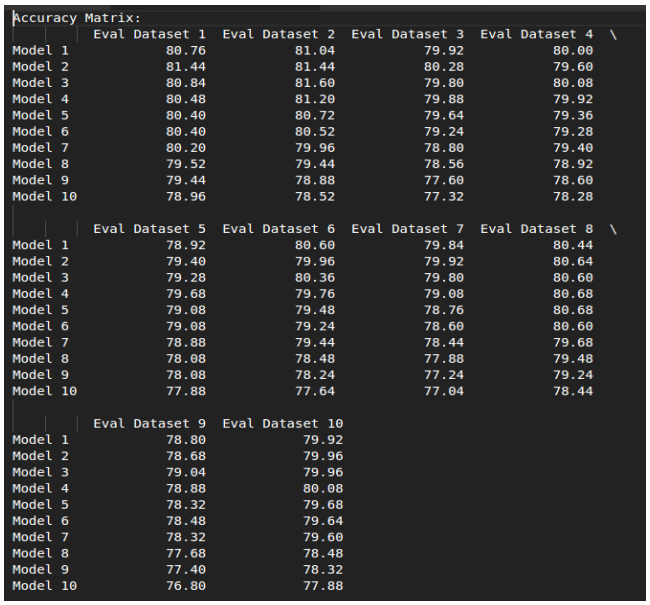
1.2.2.2 Libraries Used

- **NumPy**: For efficient array manipulation.
- **SciPy**: For generating multivariate Gaussian samples and probability calculations.
- **scikit-learn**: For `StandardScaler` and `PCA`, facilitating feature standardisation and dimensionality reduction.

This framework achieves continual learning by combining prototype-based regularisation, generative replay, and dynamic Gaussian parameter updates, ensuring adaptability and retention of knowledge across tasks.

1.2.3 Model Evaluation and Accuracy Matrix Analysis

The accuracy matrix evaluates the performance of 10 sequentially trained models across all ten evaluation datasets. Features were standardised and transformed using `StandardScaler` and PCA before evaluation. The diagonal entries reflect each model's accuracy on its corresponding dataset, while off-diagonal entries show performance on previous datasets.



```
Accuracy Matrix:
          Eval Dataset 1  Eval Dataset 2  Eval Dataset 3  Eval Dataset 4  \
Model 1            80.76           81.04           79.92           80.00
Model 2            81.44           81.44           80.28           79.60
Model 3            80.84           81.60           79.80           80.08
Model 4            80.48           81.20           79.88           79.92
Model 5            80.40           80.72           79.64           79.36
Model 6            80.40           80.52           79.24           79.28
Model 7            80.20           79.96           78.80           79.40
Model 8            79.52           79.44           78.56           78.92
Model 9            79.44           78.88           77.60           78.60
Model 10           78.96           78.52           77.32           78.28

          Eval Dataset 5  Eval Dataset 6  Eval Dataset 7  Eval Dataset 8  \
Model 1            78.92           80.60           79.84           80.44
Model 2            79.40           79.96           79.92           80.64
Model 3            79.28           80.36           79.80           80.60
Model 4            79.68           79.76           79.08           80.68
Model 5            79.08           79.48           78.76           80.68
Model 6            79.08           79.24           78.60           80.60
Model 7            78.88           79.44           78.44           79.68
Model 8            78.08           78.48           77.88           79.48
Model 9            78.08           78.24           77.24           79.24
Model 10           77.88           77.64           77.04           78.44

          Eval Dataset 9  Eval Dataset 10
Model 1            78.80            79.92
Model 2            78.68            79.96
Model 3            79.04            79.96
Model 4            78.88            80.08
Model 5            78.32            79.68
Model 6            78.48            79.64
Model 7            78.32            79.60
Model 8            77.68            78.48
Model 9            77.40            78.32
Model 10           76.80            77.88
```

Key Observations: Models maintain vital accuracy on earlier datasets, demonstrating effective knowledge retention. A gradual decline in accuracy is observed for earlier datasets with later models, reflecting the challenge of balancing adaptation and retention.

Libraries Used

- **NumPy**: For computations.
- **Pandas**: For displaying the accuracy matrix.
- **scikit-learn**: For feature preprocessing.

The results demonstrate robust continual learning with minimal forgetting.

1.3 Task 2

This task involves continually training datasets 11 to 21. We have to do Unsupervised Domain Adaptation (UDA) with a memory buff taken from each dataset using generative replay because all the datasets have dissimilar distributions. For UDA, we have to minimise the Sliced Wasserstein distance (SWD) between the already feature-extracted Memory buffer and the features extracted from the Target Dataset or current dataset using a pre-trained neural network (which is to be optimised). For this, we have to set up fine-tuning and use backpropagation to optimise, for which we will use Stochastic Gradient Descent (SGD). This implies retraining again the trained parameters or introducing new parameters to train. This becomes another deep neural network training beyond this project's scope. Also, the runtime will be exceedingly huge, beyond the scope of the machine we use (it may take days). UDA after extracting features isn't providing any improvement.

For all the reasons mentioned above, we won't be able to tackle UDA in this project, and the rest of Task 2 is the same as Task 1. We are starting with model 10 instead of LwP or its variants.

1.3.1 Model Evaluation and Accuracy Matrix Analysis

Accuracy Matrix:

	Eval Dataset 1	Eval Dataset 2	Eval Dataset 3	Eval Dataset 4
Model 11	77.76	77.36	77.12	77.48
Model 12	77.56	77.56	77.04	76.92
Model 13	76.84	77.28	75.96	76.60
Model 14	77.08	77.16	75.72	76.44
Model 15	76.44	76.56	75.44	76.56
Model 16	75.72	76.24	74.92	75.28
Model 17	75.24	76.24	75.00	75.56
Model 18	75.36	75.96	74.72	74.84
Model 19	75.08	75.72	74.64	74.72
Model 20	74.68	75.64	74.00	74.52

	Eval Dataset 5	Eval Dataset 6	Eval Dataset 7	Eval Dataset 8
Model 11	77.40	77.56	76.56	78.44
Model 12	77.16	76.96	76.04	77.36
Model 13	76.56	76.76	75.72	76.84
Model 14	76.12	76.28	75.48	77.12
Model 15	75.56	75.96	75.48	76.92
Model 16	75.24	75.28	74.84	76.64
Model 17	75.40	75.00	74.64	76.32
Model 18	75.32	74.80	75.08	76.04
Model 19	74.36	74.96	74.56	75.92
Model 20	74.36	74.36	74.20	75.52

	Eval Dataset 9	Eval Dataset 10	Eval Dataset 11	Eval Dataset 12
Model 11	76.56	77.64	55.72	41.92
Model 12	76.04	76.80	54.76	43.56
Model 13	75.20	76.20	54.28	43.16
Model 14	75.00	76.36	54.20	43.04
Model 15	74.44	76.04	53.48	42.52
Model 16	74.40	75.12	53.00	42.48
Model 17	74.04	74.68	52.96	42.28
Model 18	73.40	75.04	52.72	41.80
Model 19	73.68	74.40	52.40	41.92
Model 20	73.24	74.44	51.88	41.68

	Eval Dataset 13	Eval Dataset 14	Eval Dataset 15	Eval Dataset 16
Model 11	60.44	73.92	77.08	57.40
Model 12	60.44	73.24	76.64	57.16
Model 13	60.28	72.88	75.96	57.48
Model 14	60.16	72.56	75.92	57.04
Model 15	59.84	72.76	75.76	56.12
Model 16	59.76	72.20	75.76	57.08
Model 17	59.24	72.20	75.36	56.84
Model 18	59.56	71.64	74.60	56.60
Model 19	59.20	71.48	73.72	56.48
Model 20	58.80	71.24	73.64	55.72

	Eval Dataset 17	Eval Dataset 18	Eval Dataset 19	Eval Dataset 20
Model 11	62.80	60.88	55.64	74.40
Model 12	62.24	60.64	55.36	73.92
Model 13	61.52	60.84	55.16	73.08
Model 14	61.20	60.40	54.72	73.12
Model 15	60.80	60.00	54.84	72.68
Model 16	60.44	59.88	54.44	72.44
Model 17	61.88	59.28	54.00	72.08
Model 18	61.28	59.80	54.12	72.00
Model 19	60.64	59.64	55.80	71.76
Model 20	60.00	59.68	55.32	71.56

Since we didn't apply UDA, there are accuracy variations between evaluation datasets. However, since we have used continual learning, there are minimal accuracy variations between models.

1.4 Extracted Features

https://drive.google.com/file/d/1J4YJ7L4VXDzNUOBX_xcLgzEcr9Ey4Q2v/view?usp=sharing

2 Problem 2

https://www.youtube.com/watch?v=brLFhDZ3NY4