Expandable Subspace Ensemble (EASE) for Class-Incremental Learning: A Concise Review

Ayush Bodade
ayushbodade1@gmail.com

Abstract

- **CIL Overview:**
  - Class-Incremental Learning (CIL) enables continual learning of new classes without forgetting previously learned ones. A significant challenge in CIL is the overwriting of old knowledge when adapting to new classes, leading to catastrophic forgetting.
- **Proposed Solution:**
  - **ExpAndable Subspace Ensemble (EASE):** This framework uses lightweight adapter modules for each new task to create task-specific subspaces. Adapters span a high-dimensional feature space, facilitating joint decision-making across multiple tasks.
  - **Semantic-Guided Prototype Complement Strategy:** This strategy synthesizes new features for old classes without utilizing old class instances, addressing dimensional compatibility issues.
- **Experimental Results:**
  - Extensive testing on seven benchmark datasets demonstrates EASE's state-of-the-art performance. EASE shows competitive parameter-performance balance compared to other prompt-based methods without the need for exemplars.

1 Introduction

- **Background on Deep Learning and CIL:**
  - Deep learning has greatly improved performance in various applications. In real-world scenarios, data often arrives in streams, necessitating the incremental learning of new classes.
- **CIL Problem:**

  - Major hurdle: catastrophic forgetting, where learning new classes undermines previously learned features, known as the **stability-plasticity dilemma**.

- **Pre-Trained Models (PTMs) in CIL:**
  - The use of PTMs has gained traction as a method to improve performance in CIL due to their generalizable features from extensive datasets.

- **Limitations of Current Methods:**
  - Optimizing prompts for new tasks can conflict with those of previous tasks, resulting in catastrophic forgetting.
  - Expandable networks that keep previous models in memory and initialize new ones require significant storage and resources.

- **Challenges for CIL:**
  - Constructing low-cost, task-specific subspaces that do not interfere with old knowledge.
  - Developing classifiers that effectively map continuously expanding features to classes without relying on exemplars.

2 EASE Framework

Objective

- To mitigate cross-task conflict in continual learning (CIL) by using expandable subspaces without relying on exemplars.

Core Concepts

- **Subspace Expansion**: Create lightweight subspaces for sequential tasks to manage computational costs and maintain performance.
- **Task-Specific Features**: Adaptation modules encode features relevant to each task, preserving previously learned knowledge.
- **Classifier Synthesis**: Develop classifiers for expanding feature sets without historical instances to train on.

Adapter Mechanism

- **Adapters**: Lightweight modules attached to existing neural networks that fine-tune task-specific features.
- **Architecture**:

- Consists of a down-projection layer, a non-linear activation function, and an up-projection layer.
- Adjusts the output of MLP (Multi-Layer Perceptron) with a residual connection to retain previous knowledge.

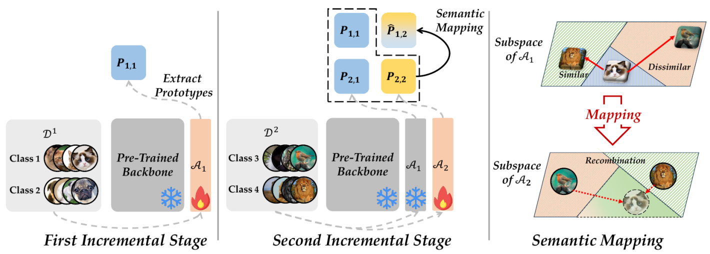

Figure 1: Architecture of the EASE Framework.

Optimization

- Pre-trained backbone weights are frozen, and only adapter parameters are optimized for each task, resulting in efficient memory usage and lower computational costs.

Final Embedding Function

- **Final Embedding Function**:

$$\Phi(x) = [\phi(x; A_1), \ldots, \phi(x; A_b)] \in \mathbb{R}^{bd} \tag{1}$$

- Combines features from all subspaces to ensure a holistic representation for classification.

3 Semantic Guided Prototype Complement

Prototype Extraction

- After training, class prototypes are extracted for each task using the embeddings produced by their respective adapters.

Prototype Reconstruction

- **Challenge**: When a new task is introduced, the model lacks exemplars to recalculate the old class prototypes in the new subspace.
- **Solution**:

- **Similarity Measurement**: Compute semantic similarity between old and new class prototypes to reconstruct old prototypes in the new subspace.
- Use a normalized similarity matrix to guide the reconstruction process, ensuring semantic relationships are maintained.
- New old class prototypes in the new subspace can be calculated as:

$$\hat{P}_{o,n}[i] = \sum_j S_{i,j} \times P_{n,n}[j] \tag{2}$$

4 Subspace Ensemble via Subspace Reweight

Full Classifier Construction

- The final classifier (prototype matrix) consists of prototypes from all tasks, including reconstructed old class prototypes.

Logit Calculation for Inference

- The classification for a task is calculated by:

$$[P_{b,1}, P_{b,2}, \ldots, P_{b,B}]^T \Phi(x) = \sum_i P_{b,i}^T \phi(x; A_i) \tag{3}$$

Reweighting Strategy

- Emphasize the contributions of the current task's prototypes during inference by adjusting their weights relative to other tasks:

$$P_{b,b}^T \phi(x; A_b) + \alpha \sum_{i \neq b} P_{b,i}^T \phi(x; A_i) \tag{4}$$

- **Trade-off Parameter**: Set to 0.1 in experiments to balance the contributions of current and previous tasks.

5 Summary of EASE Training Pipeline

- Initialize and train an adapter for each incoming task to encode task-specific information.
- Extract prototypes from the current dataset for all adapters and synthesize prototypes for previous classes.
- Construct a full classifier using synthesized prototypes and perform logit reweighting for prediction.
- **Efficiency**: The process is designed to be training-free for prototype synthesis, enhancing the efficiency of continual learning.

6 Experimental Validation

Datasets

- Conducted experiments on seven benchmark datasets: CIFAR100, CUB200, ImageNet-R, ImageNet-A, ObjectNet, OmniBenchmark, and VTAB.
- These datasets represent typical Class-Incremental Learning (CIL) benchmarks and contain large domain gaps from the pre-trained ImageNet.

Methodology

- Comparison of EASE with state-of-the-art (SOTA) algorithms to assess incremental learning ability.
- Implementation details include random shuffling of class orders for fair comparison and using ViT-B/16-IN21K as the backbone model.
- Training was performed using SGD optimizer with a batch size of 48 for 20 epochs, and the learning rate decayed from 0.01 using cosine annealing.

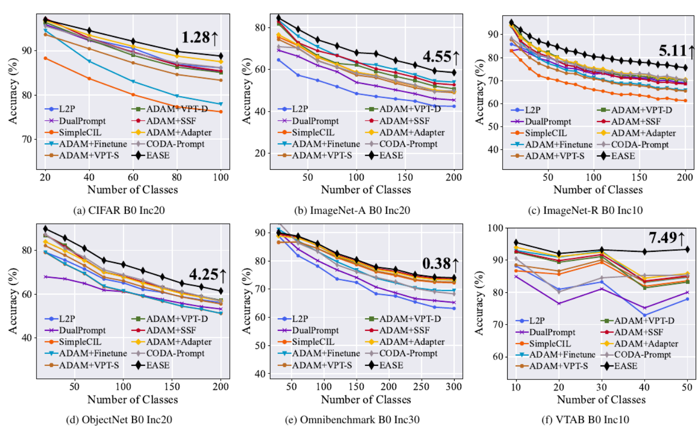

Figure 2: Experimental validation results comparing EASE with state-of-the-art methods.

Performance Comparison

- EASE outperformed all compared methods on the seven datasets, achieving the best performance among state-of-the-art methods.
- Incremental performance was analyzed against various metrics, including accuracy and the number of parameters.

7 Conclusion

- The EASE framework successfully tackles the challenge of catastrophic forgetting in class-incremental learning.
- By employing expandable subspaces and semantic-guided strategies, EASE preserves old knowledge while integrating new classes efficiently.
- Experimental results demonstrate its superiority over existing methods, showcasing its potential for real-world applications in continuous learning environments.