Review of Class-Incremental Methods

Giulia Ghisolfi, g.ghisolfi@studenti.unipi.it, 664222

Master Degree in Computer in Science (AI Curriculum), University of Pisa
Continual Learning course (791AA)
Academic Year: 2023-2024

Outline

Introduction

Incremental Learning is a paradigm where models learn new tasks continuously without retraining from scratch, reducing computational costs and reusing resources.

In **Class-Incremental Learning**, new classes are introduced incrementally at each step, older classes are not revisited, and algorithms must classify all classes encountered across time steps.

Introduction

1. **Memory-based and rebalancing classifiers:** These methods rely on memory storage or rebalancing techniques to mitigate catastrophic forgetting and address the class imbalance.
    - **iCaRL** (Incremental Classifier and Representation Learning)
    - **IL2M** (Class-Incremental Learning with Dual Memory)
    - Learning a Unified Classifier Incrementally via Rebalancing

2. **Generative approaches:** These methods represent an alternative to memory-based strategies. Instead of storing real data, they save a generative model capable of synthesizing examples of older classes during training.
    - Memory Replay GANs

Problem Setting

Challenges

Catastrophic Forgetting: New class learning overwrites prior knowledge, requiring regularization or loss approximation to preserve past classes.

Stability-Plasticity Trade-Off: Balancing retention of old knowledge (stability) and adaptation to new classes (plasticity).

Class Imbalance: Recent classes dominate predictions, leading to *classifier bias*.

Knowledge Distillation

Tecnique to transfers knowledge from a larger (*teacher*) to a smaller (*student*) model.

Mitigates forgetting by applying distillation to old classes and Cross-Entropy loss to all classes, addressing mismatched output distributions as the classifier grows.

Methodology: Class-Incremental Classifier

Incremental Classifier and Representation Learning

Combines knowledge representation and knowledge distillation to address catastrophic forgetting, class imbalance, and effectively manage a large number of classes over time.

Prototype-based Strategy (Nearest-Mean-of-Exemplars): Classifies inputs by finding the nearest class prototype in feature space.

- Class prototype (μ_y) is the mean feature vector of exemplars: $\mu_y = \frac{1}{|P_y|} \sum_{p \in P_y} \phi(p)$.
- Classify x by finding the nearest prototype: $y^* = \arg\min_{y=1,\ldots,t} \|\phi(x) - \mu_y\|$.

Representation Learning with Knowledge Distillation: Combines new and exemplar data to preserve old class knowledge while learning new ones.

- **Classification Loss:** Encourages accurate predictions for new classes.
- **Distillation Loss:** Preserves old class knowledge.

$$\mathcal{L}(\Theta) = \mathcal{L}_{\text{classification}} + \mathcal{L}_{\text{distillation}}.$$

Exemplar Selection (Herding): Maintains a fixed memory size by selecting the most representative samples for each class.

- Allocates $m = \frac{K}{t}$ exemplars per class.
- For new classes, selects exemplars to minimize deviation from class mean: $p_k = \arg\min_{x \in X} \left\| \mu - \frac{1}{k} \left(\phi(x) + \sum_{j=1}^{k-1} \phi(p_j) \right) \right\|$.

Class-Incremental Learning with Dual Memory

Builds on the idea that classes are best modeled when first learned with all data available.

Introduces a dual memory structure:

- **First Memory:** Stores exemplar images of past classes as a representative subset.
- **Second Memory:** Stores initial class statistics, providing optimal representations with minimal storage cost.

Employs a distillation-based loss to align predictions and mitigate catastrophic forgetting.

Bounded Memory and Data Imbalance:

- Retains $m = \frac{K}{t}$ exemplars per class.
- Imbalance increases as classes grow.

Prediction Rectification: To correct biases toward new classes, predictions for past classes C_i are rectified using:

$$p^r(x) = \begin{cases} p(C_i) \times \frac{\mu(M_N)}{\mu(M_P)} \times \frac{\mu_N(C_i)}{\mu_P(C_i)}, & \text{if pred} = \text{new}, \\ p(C_i), & \text{otherwise.} \end{cases}$$

Efficiency: The rectification formula adds minimal computational overhead, maintaining the method's efficiency while reducing bias toward new classes.

Learning a Unified Classifier Incrementally via Rebalancing

Incrementally learn a unified classifier that systematically investigates and addresses the effects of class imbalance, improving training balance and reducing bias toward new classes.

Cosine Normalization: Ensures balanced predictions with normalized weight vectors:

$$p(x) = \frac{\exp(\eta \cdot \langle \bar{\theta}_i, \bar{f}(x) \rangle)}{\sum_j \exp(\eta \cdot \langle \bar{\theta}_j, \bar{f}(x) \rangle)}$$

Learning a Unified Classifier Incrementally via Rebalancing

Less-forget Constraint: Preserves the geometric configuration of old classes using distillation loss:

$$L_{\text{LF}} = 1 - \langle \bar{f}^*(x), \bar{f}(x) \rangle,$$

weighted by $\lambda = \lambda_{\text{base}} \cdot \frac{|C_n|}{|C_o|}$.

Inter-class Separation: Enhances separation between old and new classes using margin-based loss:

$$L_{\text{margin}} = \sum_{k=1}^{K} \max\left(0, m - \langle \bar{\theta}(x), \bar{f}(x) \rangle + \langle \bar{\theta}_k, \bar{f}(x) \rangle\right).$$

Final Loss: $L = \frac{1}{|N|} \sum_{x \in N} (L_{\text{ce}}(x) + \lambda L_{\text{LF}}(x)) + \frac{1}{|N_o|} \sum_{x \in N_o} L_{\text{margin}}(x)$.

Methodology: Generative Replay

Memory Replay GANs

Address sequential learning by incorporating a conditional GAN framework with three components:

- **Generator (G):** Synthesizes data for both old and new classes.
- **Discriminator (D):** Distinguishes real from generated images.
- **Classifier (C):** Predicts labels, forcing G to generate meaningful images aligned with target labels.

Data conditioned on class label c and latent vector z: $\tilde{x} = G(z, c)$.

Model addresses both adversarial and classification tasks during training.

Memory Replay GANs

Sequential Learning in GANs: GAN training extended for incremental learning with parameters initialized from the previous task:

$$\min_{\theta_G^t} \mathcal{L}_{\mathrm{GAN}}^{G}(\theta_t, S_t), \quad \min_{\theta_D^t} \mathcal{L}_{\mathrm{GAN}}^{D}(\theta_t, S_t),$$

where $\theta_t = (\theta_G^t, \theta_D^t)$.

Elastic Weight Consolidation (EWC): Penalizes significant changes in important parameters to prevent forgetting:

$$\min_{\theta_G^t} \mathcal{L}_{\mathrm{GAN}}(\theta_G^t, S_t) + \frac{\lambda_{\mathrm{EWC}}}{2} \sum_i F_{t-1,i}(\theta_{G,t,i} - \theta_{G,t-1,i})^2.$$

Memory Replay Mechanism: *Joint Retraining with Replay:* Combines synthetic data from old tasks with real data from the current task for training. *Replay Alignment:* Synchronizes G_t and G_{t-1} to generate identical outputs for a given z, c.

Analysis and Comparison between Methods

Class-Incremental Learning Classifiers Comparison

Common Features: All methods use memory buffers, distillation techniques, and address class imbalance.

Differences:

- **iCaRL**: Nearest-mean classifier with fixed memory. Does not explicitly handle class imbalance, leading to lower accuracy for old classes.
- **IL2M**: Dual memory (exemplars + class statistics). Rectifies old class predictions, improving accuracy with low memory usage.
- **LUCIR**: Combines advanced techniques (cosine normalization, less-forget constraint, inter-class separation), to explicitly balance training and reduce bias toward new classes.

Experimental Results

iCaRL vs IL2M: IL2M consistently outperforms iCaRL across all memory sizes and datasets (ILSVRC, VGGFace2, Landmarks), demonstrating superior accuracy and robustness, particularly with limited memory.

iCaRL vs LUCIR: The Unified Classifier improves accuracy by 6% on CIFAR-100 and reduces error by 13% on ImageNet compared to iCaRL, effectively balancing training and reducing bias toward new classes.

Experimental Results

MeRGANs reduce catastrophic forgetting by generating high-quality samples of old classes, providing a scalable, memory-efficient alternative for class-incremental learning. However, further improvements are needed for complex datasets like CIFAR-100.

Authors also demonstrate that the generated images provide a visual representation of task interference and potential forgetting.

Strengths and Weaknesses

Strengths and Weaknesses

Method	Strengths	Weaknesses
iCaRL	Simple and effective nearest-mean classifier.	Lacks mechanisms for class imbalance.
	Knowledge Distillation preserves knowledge of old classes.	Limited flexibility due to fixed-size memory.
IL2M	Two-memory structure corrects predictions.	Limited scalability due to compact statistics.
	Low computational overhead.	No explicit optimization of feature space.
	Strong theoretical justification for statistics.	
LUCIR	Combines techniques to address imbalance and forgetting.	Higher computational complexity.
	Explicitly balances training.	Requires careful hyperparameter tuning.
MeRGANs	Generative replay reduces memory requirements.	Instability during GAN training.
	Maintains old/new class classification.	Computationally expensive.
	Visual representation of task interference and forgetting.	Does not address feature space separation.

Reference

E. Belouadah and A. Popescu.
Il2m: Class incremental learning with dual memory.
In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 583–592, 2019.

S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin.
Learning a unified classifier incrementally via rebalancing.
In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 831–839, 2019.

S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert.
icarl: Incremental classifier and representation learning.
In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

C. Wu, L. Herranz, X. Liu, J. Van De Weijer, B. Raducanu, et al.
Memory replay gans: Learning to generate new categories without forgetting.
Advances in neural information processing systems, 31, 2018.