BTech Thesis Presentation

Online Continual Learning

Chirag
12140520

Indian Institute of Technology Bhilai

May 2, 2024

Presentation Outline

Introduction & Motivation

Introduction

- The current dominant paradigm for ML is to run an ML algorithm on a given dataset to generate a model. The model is then applied in real-life performance tasks. This is called isolated learning because it does not consider any other related information or the previously learned knowledge.
- The fundamental problem with this isolated learning paradigm is that it does not retain and accumulate knowledge learned in the past and use it in future learning.

Motivation

- In contrast, we humans learn quite differently. We accumulate and maintain the knowledge learned from previous tasks and use it seamlessly in learning new tasks and solving new problems.
- When faced with a new problem or a new environment, we can adapt our past knowledge to deal with the new situation and also learn from it. Over time we learn more and more, and become more and more knowledgeable and more and more effective at learning.

Problem Statement

- The problem of Online Continual Learning (OCL) is characterized by the need for machine learning models to adapt to evolving data distributions over time, often encountered in dynamic environments such as streaming data, online platforms, or IoT devices.
- Requires the strategies to efficiently incorporate new information while preventing detrimental interference with existing knowledge, necessitating the development of algorithms that strike a balance between plasticity and stability, ensuring continual improvement without sacrificing past learning experiences.

Challenges

Challenges

- Catastrophic forgetting: Models tend to forget previously learned knowledge when exposed to new data, leading to performance degradation.
- Task interference: Features learned for one task may interfere with those needed for another task.
- Scalability: Managing a large number of tasks while avoiding catastrophic forgetting requires efficient algorithms and architectures.
- Resource constraints: Continual learning requires significant computational resources, including memory and processing power.
- Data efficiency: Efficiently learn from limited data for new tasks while preserving knowledge from previous experiences.
- Knowledge transfer: Efficient methods for transferring previously learned knowledge to new tasks, enhancing model performance and efficiency.

Research Work

In addressing the challenges of continual learning, researchers have explored three main approaches: Parameter Isolation, Memory Based Replay, and Regularization

- Parameter Isolation: Isolating critical parameters related to previous tasks and preventing them from being updated during training on new tasks, preserving important knowledge.
- Memory Based Replay: Representative experiences from past tasks are stored in a memory buffer. During training on new tasks, the model retrieves and replays these experiences to retain knowledge and prevent catastrophic forgetting.
- Regularization: Introduce constraints on parameter updates during training on new tasks. Methods like Elastic Weight Consolidation (EWC) penalize significant changes to important parameters learned from previous tasks, helping the model retain valuable knowledge while adapting to new information.

Research Gap/Potential Improvements

Research Gap

- Parameter Isolation often leads to inefficient use of model capacity and a rigid architecture that limits the ability to leverage commonalities across tasks, hindering performance in environments where task generalization is crucial.
- Memory-Based Replay faces challenges related to privacy, security, and scalability, struggling to maintain performance as tasks grow in number and data diversity, especially where data storage is impractical.
- Regularization Techniques like Elastic Weight Consolidation can be overly conservative, unnecessarily constraining the learning of new tasks and impeding rapid adaptation in dynamic settings.

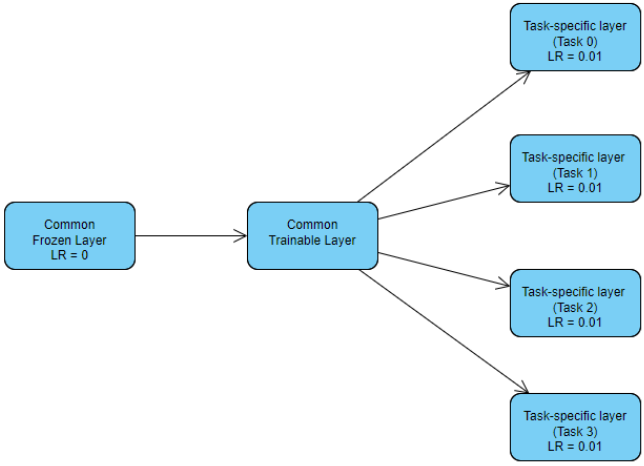
These limitations highlight the need for more adaptable, flexible architectural designs and dynamic learning strategies that can effectively balance knowledge retention with the efficient acquisition of new information.

Our Approach

Our Approach

- Multi-headed Resnet-18 architecture with three types of layers: Common Frozen Layers, Common Trainable Layers & Task-Specific Layers/Heads.
- Common Trainable Layers are the set of parameters common to all the heads and will be updated over each training step along-with Task-Specific Layers.
- Idea behind the common trainable layer is to study the effect of sharing weights across tasks and their impact on catestrophic forgetting.
- Strategic Learning Rates: Differentiated learning rates were applied, with lower rates for common trainable layers to prevent drastic updates that might lead to forgetting, and higher rates for task-specific layers to promote quick adaptation to new challenges.

Our Approach



Detailed Experimental Setup

Detailed Experimental Settings (Contd.)

- Backbone Architecture: Resnet-18 Pre-trained.
- Datasets: MNIST, CIFAR10, CIFAR100, FMNIST.
- Splits for Common Frozen Layers, Common Trainable Layers and Task-Specific Layers (in Millions):
  - 3-3.5-4.8
  - 3-1-7.2
  - 4-2.5-4.8
  - 6.5-2.3-2.4
  - 4-4.8-2.4
- Batch-size: 256, 64 (Setting 1) and 512 (Setting 2)
- Learning Rate: 0.000001 to 0.01 (Common Trainable Layer), 0.01 (Task-Specific Layer)

Detailed Experimental Settings (Contd.)

Experimental Settings

- Setting 1: Pass entire MNIST data then CIFAR10, CIFAR100 and FMNIST data respectively while training.
- Setting 2: Merge all the training datasets and pass the batch corresponding to any of the dataset randomly (Only for those splits which out-performed in Setting 1)

Results

Results (Setting 1)

LR (Common-trainable layer) = 0.00001 for all kind of splits. For top performers (6.5-2.3-2.4 & 4-4.8-2.4), further experiments are done.

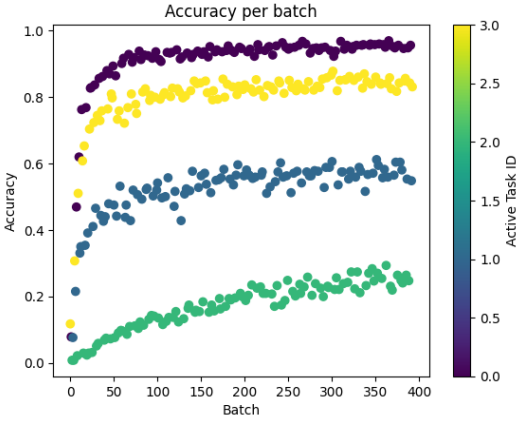
Split (in M)	Dataset	B1-Val1	B1-Val2	B2-Val1	B2-Val2	B3-Val1	B3-Val2
3-3.5-4.8	MNIST	98	89	99	81	99	48
3-3.5-4.8	CIFAR10	68	45	68	39	69	27
3-3.5-4.8	CIFAR100	35	30	39	33	39	34
3-3.5-4.8	FMNIST	89	89	90	90	90	90
3-1-7.2	MNIST	98	75	99	65	99	31
3-1-7.2	CIFAR10	68	36	69	29	68	29
3-1-7.2	CIFAR100	36	31	39	33	38	29
3-1-7.2	FMNIST	89	89	91	91	90	90
4-2.5-4.8	MNIST	98	89	99	60	99	60
4-2.5-4.8	CIFAR10	67	28	68	23	68	32
4-2.5-4.8	CIFAR100	32	27	38	31	38	32
4-2.5-4.8	FMNIST	89	89	90	90	90	90

Split (in M)	Dataset	B1-Val1	B1-Val2	B2-Val1	B2-Val2	B3-Val1	B3-Val2
6.5-2.3-2.4	MNIST	96	96	97	96	97	96
6.5-2.3-2.4	CIFAR10	62	55	63	50	62	52
6.5-2.3-2.4	CIFAR100	32	32	37	36	36	35
6.5-2.3-2.4	FMNIST	86	86	87	87	87	87
4-4.8-2.4	MNIST	96	95	97	93	97	94
4-4.8-2.4	CIFAR10	59	55	60	52	61	50
4-4.8-2.4	CIFAR100	29	28	32	32	34	33
4-4.8-2.4	FMNIST	85	85	86	86	86	86

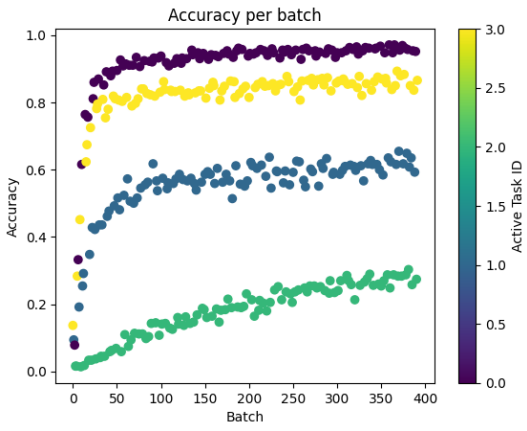
Results (Setting 1) Contd..

- After changing various learning rates and batch-sizes for splits 6.5-2.3-2.4 & 4-4.8-2.4, the optimal learning rate, split and batch-size is figured out.
- B1 = 256, B2 = 64
- Val1: Validation Accuracy when passed for the first time.
- Val2: Validation Accuracy when all the datasets are passed.

Results (Setting 2)



4-4.8-2.4 split



6.5-2.3-2.4 split

Results (Setting 1) Contd..

- Forgetting and Stability: The model exhibited varying degrees of forgetting, particularly noticeable in smaller batch sizes and higher learning rates. The configurations with larger batch sizes generally provided more stable learning outcomes.
- Setting 1 vs. Setting 2: Setting 1, which involved sequential task learning, showed better performance and less forgetting compared to Setting 2, where tasks were learned from randomly sampled batches. This highlighted challenges in adapting to non-sequential and mixed-task environments.
- Top Performers: The optimal configurations, notably the 6.5-2.3-2.4 and 4-4.8-2.4 splits, balanced high accuracy with minimal forgetting across tasks, indicating their effectiveness in managing the trade-offs inherent in continual learning.

Analysis

Analysis

- For lower batch-size, high learning rate leads to high catastrophic forgetting.
- Task-specific layers should be as thin as possible (High forgetting in case of bigger heads)
- For larger batch-size, learning rate for the common trainable layer doesn't really matter (returns similar results for both the splits)
- Accuracy in Setting 2 is lesser than that in Setting 1 (for example, for task2, accuracy in setting 2 is $\sim$20-25% while in setting 1, it's $\sim$30-34%)
- Best split (high accuracy and low forgetting with least number of trainable params) is 6.5-2.3-2.4 for these datasets.

Future Works

- Testing of our approach on real-world datasets (Skin disease and WILDS datasets)
- Automated detection algorithm for identifying the task the batch belongs to.
- Task-dependent splits and architectures for task-specific layers.
- Mixture of Experts-based architecture for merging outputs of multiple heads at a time for better learning and inference.

“If you have knowledge, let others light their candles in it.”

Margaret Fuller

Thank You!