# NTU CE7454 Project 1: Class-Incremental Learning Challenge

Seow Wei Liang
Nanyang Technological University
50 Nanyang Avenue, Singapore 639798
WEILIANG003@e.ntu.edu.sg

## Abstract

*The aim of this project is to perform class-incremental learning using a data subset of the original OmniBenchmark. A SimpleCIL model utilizing pre-trained visual transformer model (pretrained_vit_b16_224_in21k) was fine tune for each task for class-incremental learning. Various optimization techniques were used to improve the empirical loss such as optimizing the learning rate and dataset normalization. Regularization techniques were used to improve the test loss which include the weight decay, LoRa dropout, data augmentation techniques(RandomResizedCrop, Random Horizontal flip and Augmix) and label smoothing. To tackle class imbalance problem in class-incremental learning, Weighted Random Sampling was used to sample data batches using class weights. Focal loss was also explored to improve performance for class imbalanced learning. Various techniques were employed to prevent catastrophic forgetting. Low-Rank Adaptation of Large Language Models (LoRA) was used to freeze most of the network parameters during fine tuning to reduce catastrophic forgetting. To further retain the knowledge learnt from the previous task, weight averaging was also performed by using the previous model and the model from the current iteration. Elastic weight consolidation was also explored to prevent catastrophic forgetting by constraining the weights important from the previous task.*

## 1. Introduction

This project focuses on Class Incremental Learning which learns new classes in the current task without forgetting the old classes from previous task. During the learning of new classes from new task, data about the old classes from previous tasks are unavailable. A good trained model is then required to learn the new classes from the current task without forgetting the old classes learnt from previous tasks. In the event that the model forgets most of the knowledge learnt from previous tasks, this phenomenon is known as catastrophic forgetting. Various techniques can alleviate catastrophic forgetting such as Low-Rank Adaptation of Large Language Models (LoRA) [3], Weight Averaging [1] and Elastic Weight Consolidation (EWC) [5]. In this paper, we will use the SimpleCIL model [7] which continually sets the classifiers of pretrained model(PTM) to prototype features to study the optimization, regularization during fine tuning for every tasks and also perform an ablation study to the important techniques implemented.

Different optimization techniques were explored to improve the empirical loss which include optimizing the learning rate, dataset normalization. Regularization techniques were explored to improve the test loss which include the weight decay, LoRa dropout, data augmentation techniques(RandomResizedCrop, Random Horizontal flip and Augmix [2]) and label smoothing. To tackle the class imbalance problem in class-incremental learning, Weighted Random Sampling and focal loss [6] were also explored to improve performance. We have also done hyperparameter tuning to study the effect on learning rate on the average top 1 accuracy and present the effect of different weight decay and dropout on the average top 1 accuracy.

We will also conduct an ablation study to study the effects of different techniques such as EWC, weight averaging, LoRa and focal loss.

## 2. Methodology

Methodology used in this project includes optimization techniques, regularization techniques, class imbalance techniques and continual learning methods used to prevent catastrophic forgetting.

### 2.1. Optimization techniques

Optimization techniques were used to improve the empirical loss on the training dataset. The most critical is tuning the learning rate which controls the learning at each gradient descent step. SGD optimizer was chosen as it provides the better generalization compared to Adam [4] optimizer. Although Adam optimizer provides adaptive learning rate, the learning rate is small nearer optima which causes Adam to accidentally accelerate the learning rate and overshoot

the global minimum. A learning rate scheduler is also used which is the cosine learning rate annealing which decay the learning rate based on the cosine curve to a minimum learning rate so as not to overshoot at the global minimum.

Data whitening was utilized to improve optimization during learning. By calculating dataset mean and standard deviation for each task, images are normalized to zero mean and variance one during training. This can lower the condition number of the feature covariance matrix and improves the speed of convergence.

### 2.2. Regularization techniques

Regularization techniques are used to prevent model overfitting. Overfitting is a phenomenon occurred during model training when the machine learning model gives accurate predictions for training dataset but not validation dataset. Several regularization methods were used include data augmentation, weight decay, dropout and label smoothing can help alleviate overfitting.

Data augmentation methods used in training include RandomResizedCrop, RandomHorizontalFlip and AugMix [2]. AugMix uses mixing the results from augmentation chains or compositions of augmentation operations from AutoAugment.

Weight decay (L2 regularization) utilizes the squared magnitude of the coefficient as the penalty term to the loss function. This helps to regularize and make the network simpler thus reducing overfitting.

Dropout is also used to regularize the network similar to L2 regularization but does it in a different way. This technique randomly sets network neurons to zero with a defined probability during model training.

Focal loss with label smoothing was also explored as a regularization technique. Label Smoothing introduces noise in the labels which accounts for labelling errors in training dataset.

### 2.3. Class imbalanced techniques

Class imbalance occurs when there is data shortage for certain classes and surplus of data from other classes. This creates dataset class imbalance and during backpropogation the gradient of the dominant class dominates the entire gradient for backpropogation. There are two main techniques to tackle class imbalance which is used in this paper. One of them is Weighted Random Sampling which utilize the class weights in the train dataloader to perform Weighted Random Sampling of data batches. The class weights are computed as using a compute_class_weight function from sklearn library with class weights set to balanced. Another method used in this paper is focal loss [6] which rapid lowers the loss for the easy examples and focus the training on the hard examples. Focal loss as shown in equation 1 has two hyperparameters $\alpha$ and $\gamma$. $\alpha$ is the class weights of each label which has similar meaning to the one in the Weighted Random Sampler. $\gamma$ controls how much to penalize the loss if the model is confident of the predictions. For this paper, we use $\alpha$ of 1 because we have utilized weighted random sampling and $\gamma$ of 1. $\gamma$ more than 1 will penalize the loss too much such that it inhibits learning of easy examples while $\gamma$ less than 1 does not allow the model to focus much on the hard examples. As a note, Focal loss with $\alpha$ of 1 and $\gamma$ of 0 is equivalent to cross entropy loss.

$$FL(p_t) = -\alpha(1-p_t)^{\gamma} log(p_t) \tag{1}$$

### 2.4. Continual learning techniques

Three techniques were used for continual learning to prevent catastrophic forgetting. The first method is Low-Rank Adaptation technique (LoRa) [3], a parameter efficient fine tuning method. LoRa, freezes the pre-trained model weights and injects trainable rank decomposition matrices into LoRa target layers of the Visual Transformer, massively reducing the number of trainable parameters during fine tuning. This provides two main benefits by enabling larger batch sizes model fine tuning and minimize catastrophic forgetting as majority of the model parameters were freezed during fine tuning.

Another method that is useful in retaining knowledge during continual learning is Weight Averaging [1]. Weight Averaging is a simple, easy to implement method to prevent catastrophic forgetting by taking the average of the weights from the previous model $\theta^{t-1}$ and trained model $\hat{\theta}^t$ from the current iteration as shown in equation 2. $\eta$ can be 0.5 where we consider equal contributions from $\theta^{t-1}$ and $\hat{\theta}^t$ to calculate the new weights $\theta^t$ or $\eta$ can be calculated based on the known classes versus all classes at the current iteration. In this paper, we utilized the latter.

$$\theta^t = (1-\eta)\theta^{t-1} + \eta\hat{\theta}^t \tag{2}$$

The last method involves Elastic Weight Consolidation [5]. Elastic Weight Consolidation (EWC) selectively decreasing weights plasticity, allowing knowledge of previous tasks to be retained during learning of new tasks and hence alleviating catastrophic forgetting. This method calculates the fisher information matrix $F_i$ by performing a single pass through the old task training set and performing the gradient calculation of each weights. Using $F_i$ together with the previous task weights $\theta_{A,i}$ and current weights $\theta_i$, we implement EWC as a soft, quadratic constraint in addition to original focal loss function $L_B(\theta)$ by pulling back each weight in new task towards its old values from previous tasks proportional to its importance $\lambda$ as shown in equation 3.

$$L(\theta) = L_B(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta^*_{A,i})^2 \tag{3}$$

## 3. Experiments

In this section, we will explain the dataset used, implementation details, discuss the effects of different optimization and regularization hyperparameters on our model performance, perform an ablation study to observe the effects of different techniques used.

### 3.1. Dataset

The dataset used in this challenge is a subset of OmniBenchmark, which has 650,447 images across 5,719 classes. Only 300 class are utilized in this challenge. For every new task, 30 new classes were added on top of the previous classes for evaluation.

### 3.2. Implementation details

SimpleCIL model was implemented with fine tuning for each task based on the pre-trained visual transformer model (pretrained_vit_b16_224_in21k).

Data was sampled for training using weighted random sampling and we calculate the class weights for selection of samples from each class. Auto mixed precision was utilized to save memory and allow larger batch sizes. We use an SGD optimizer with learning rate of 0.025 and momentum of 0.9, 15 epochs, batch size of 32 for the task 0. Learning rate of 5e-8, 2.5e-4, 7e-5, 5e-7, 4e-7, 2e-7, 1e-7, 1e-7, 1e-7 with momentum of 0.9 was utilized for task 1 to task 9 respectively. For task 1 to task 9, 1 epoch and batch size of 32 was used. Weight decay of 5e-4 was used for each task. A learning rate scheduler CosineAnnealingLR was used to decay the learning rate as the training occurs with minimum learning rate of 1e-8. Data whitening was performed with separate mean and standard deviation calculated for each task. Data augmentation methods used in training include RandomResizedCrop of scale of (0.05, 1.0) and ratio of (3/4, 4/3), RandomHorizontalFlip with probability of 0.5 and AugMix [2]. Augmix uses the default hyperparameters in the original implementation. Focal loss with alpha=1 and gamma=1 was used for all tasks. Elastic weight consolidation (EWC) loss was added to the focal loss and minimise together as a combined loss. For LoRA implementation, the key hyperparameters used are lora_dropout=0.2, lora_alpha=16, r=16 and target_modules are the attention qkv layers used to construct the LoRa layers. Weight averaging is done after fine tuning of each task and before replacing the fully connected (fc) layer by its encoder trained features. Focal loss with label smoothing was also explored with smoothing factor of 0.1.

### 3.3. Effect of Learning Rate

In this section, we used the SimpleCIL fine tune model with hyperparameters described in implementation details and tweak the task 0 learning rates. We explored different task 0 learning rates of 0.015, 0.025, 0.035 and record the average top 1 accuracy. Learning rate of 0.025 gives the best result as shown in the table 1 which obtain the highest average top 1 accuracy of 81.219. Having too low a learning rate will make the model reach the optima slower. Having too high a learning rate may cause model divergence during learning and overshoot near the optima.

| Task 0 Learning Rate | Average Top 1 accuracy |
|---|---|
| 0.015 | 80.816 |
| 0.025 | 81.219 |
| 0.035 | 80.429 |

Table 1. Effect of different task 0 learning rate on the Average Top 1 accuracy

### 3.4. Effect of Weight Decay

In this section, we used the SimpleCIL fine tune model with hyperparameters described in implementation details and tweak the weight decay for SGD optimizer. We explored different weight decay of 5e-3, 5e-4, 5e-5 for all tasks and record the average top 1 accuracy across all tasks. AS shown in table 2, Weight decay of 5e-4 gives the best result as shown in the table which obtain the highest top 1 accuracy for all tasks. Having too high a weight decay may result in the model having smaller weights which are unable to give a good accuracy. Conversely, too low a weight decay does not provide the model with sufficient regularization and resulting in model overfitting.

| Weight decay | Average Top 1 accuracy |
|---|---|
| 5e-3 | 78.242 |
| 5e-4 | 81.219 |
| 5e-5 | 80.987 |

Table 2. Effect of weight decay on the Average Top 1 accuracy

### 3.5. Effect of LoRa Dropout

In this section, we used the SimpleCIL fine tune model with hyperparameters described in implementation details and tweak the LoRa dropout probability. We explored different LoRa dropout of 0.1, 0.2, 0.3 for all tasks and record the top 1 accuracy of all tasks. As shown in table 4, LoRA dropout of 0.2 gives the best result as shown in the table which obtain the highest top 1 accuracy for all tasks. Having too high a dropout may result in the model having too little nodes activated resulting in a simple model with lower accuracy. Conversely, too low a dropout does not provide the model with sufficient regularization resulting in model overfitting.

| LoRa dropout | Average Top 1 accuracy |
|---|---|
| 0.1 | 80.799 |
| 0.2 | 81.219 |
| 0.3 | 80.614 |

Table 3. Effect of LoRa dropout on the Average Top 1 accuracy

### 3.6. Effect of Label Smoothing

In this section, we used the SimpleCIL fine tune model with hyperparameters described in implementation details and experiment with label smoothing with smoothing factor of 0.1. Specifically, we compared the focal loss with smoothing and focal loss. Focal loss with smoothing perform worse at 80.552 compared to focal loss of 81.219. This can be attributed to clean dataset. If noisy dataset was used, focal loss with smoothing will produce better performance.

| Loss function | Average Top 1 accuracy |
|---|---|
| Focal loss with smoothing | 80.552 |
| Focal loss | 81.219 |

Table 4. Effect of label smoothing on the Average Top 1 accuracy

### 3.7. Ablation study

In this section, we used the SimpleCIL fine tune model with hyperparameters described in implementation details and performed an ablation study by removing the implementation of EWC, weight averaging, focal loss and LoRa incrementally to study the effects on the top 1 accuracy as shown in table 5. With the removal of Elastic Weight Consolidation (EWC), accuracy drops from 81.219 to 81.075. Removing weight averaging results in accuracy to drop to 81.067 from 81.075 which is a smaller drop which shows that EWC has a greater impact than weight averaging in reducing catastrophic forgetting. Removing focal loss and replace with cross entropy loss led to further drop in top 1 accuracy from 81.067 to 79.815. This shows that focal loss indeed help to reduce the impact of class imbalance where classes from previous tasks were unavailable for training. Removing LoRa caused average top 1 accuracy to drop sharply from 79.815 to 39.435. This shows that LoRa which freeze most of the network parameters during fine tuning and only fine tune the LoRa layers plays a major role in preventing catastrophic forgetting.

## 4. Conclusion

In summary, we have performed optimization through learning rate and regularization through weight decay, data augmentation, dropout, label smoothing on the SimpleCIL fine tune model for each task and also explored various techniques to prevent catastrophic forgetting such as LoRa, EWC, weight averaging. We also explored ways to reduce the effect of class imbalance such as Weighted Random Sampling, focal loss. In addition, we also performed an ablation study on the key techniques implemented.

| Techniques | Average Top 1 accuracy |
|---|---|
| All | 81.219 |
| All-EWC | 81.075 |
| All-EWC -Weight Averaging | 81.067 |
| All-EWC -Weight Averaging -Focal loss | 79.815 |
| All-EWC -Weight Averaging -Focal loss -LoRa | 39.435 |

Table 5. Ablation study for different techniques on the Average top 1 accuracy

## 5. Acknowledgement

I would like to thank Prof Li Boyang, Albert and Prof Liu Ziwei for organizing this class incremental challenge.

## References

[1] Steven Vander Eeckt and Hugo Van hamme. Weight averaging: A simple yet effective method to overcome catastrophic forgetting in automatic speech recognition, 2023. 1, 2

[2] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty, 2020. 1, 2, 3

[3] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. 1, 2

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 1

[5] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 1, 2

[6] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. 1, 2

[7] Da-Wei Zhou, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need, 2023. 1