Few-shot Class Incremental Learning via Subspace Regularization with Reusing Novel-Class Weights (Novel 클래스의 가중치를 재사용하는 부분 공간 규제를 통한 소수샷 클래스 증강 학습)

요약

Few-shot Class Incremental Learning (FSCIL)은 적은 양의 데이터 (few-shot)으로 하나의 모델을 여러 태스크에 걸쳐 학습시키는 (class incremetnal) 방식이다. 적은 양의 데이터를 사용한 모델의 학습은 오버피팅을 야기하기 쉽고, 여러 태스크에 걸친 모델의 학습은 기존 태스크에서 학습한 지식을 잊어버리는 파괴적 망각(catastrophic forgetting)이 발생하기 쉽다. 기존 연구에서는 이러한 문제를 해결하기 위해 부분 공간을 통한 규제를 제안했지만 base 클래스의 수가 적어지는 경우 성능이 하락하게 된다. 이에 본 연구에서는 이전에 학습된 novel 클래스의 가중치를 재사용하여 base 클래스의 수가 적은 상황에서도 보다 나은 성능을 보이는 방법을 제시한다. 또한, 소스코드를 GitHub 에 공개하여 향후 연구에 도움이 되고자 한다.

1. 서론

1.1. 연구배경

현대의 딥러닝 기술은 컴퓨터 비전과 자연어 처리, 강화 학습 등 수많은 분야에서 매우 높은 성능을 보이고 있다. 그 중, 컴퓨터 비전 분야에서는 ImageNet [1] classification 문제에서 top-1 accuracy 90.94%를 달성하는 등 [2] 아주 높은 성능을 보이고 있다.

그러나 이러한 높은 성과는 현실 세계의 문제를 해결하기에는 다소 동떨어진 환경에서 발생한 것이다. 예시로 들었던 ImageNet [1] 의 경우, 1000 개의 클래스에 걸쳐 약 130 만장의 이미지를 사용하는데, 현실세계에서 이런 데이터는 수집하고 클래스를 표기하는 데 수많은 비용이 든다.

또한, 기존의 딥러닝 기술들은 대개 하나의 모델이 하나의 태스크 혹은 데이터셋을 학습하게 되는데, 새로운 데이터가 추가되는 경우 모델을 새로 학습해야 하기 때문에 비효율적이다.

위의 두 문제를 해결하기 위해 최근의 딥러닝 기술은 few-shot learning 과 lifelong learning 을 주제로 활발히 연구되고 있다. Few-shot learning 은 데이터가 적은 환경에서 모델을 효과적으로 학습하기 위한 기술이다. 대개 현존하는 딥러닝 기술들은 많은 데이터를 가정하고 있는데, 적은

수의 데이터가 주어지게 되면 오버피팅(over-fitting) 현상이 발생하게 된다.

한편, lifelong learning 은 하나의 모델이 여러 개의 태스크를 순차적으로 학습시키기 위한 방법을 연구한다. 전통적인 딥러닝 방법은 새로운 태스크를 학습하기 위해 fine-tuning 방법을 적용하는데, 이러한 경우 새로운 태스크를 학습하게 되면 기존의 태스크에 대한 지식은 잊어버리는 catastrophic forgetting 이 발생하게 된다. Lifelong learning 은 태스크를 명시하는 task incremental learning 과 태스크를 명시하지 않는 class incremental learning 으로 나눌 수 있다.

1.2. 연구목표

본 연구에서는 앞서 언급했던 few-shot learning 과 lifelong learning 중 class incremental learning 을 조합하여 적은 데이터로도, 여러 개의 태스크를 효과적으로 학습할 수 있는 few-shot class incremental learning (FSCIL)에 대해 연구하고자 한다.

2. 관련 연구

2.1. Few-shot learning

Few-shot learning 은 적은 수의 학습 데이터로도 모델이 좋은 성능을 내도록 하는 학습 방법이다. Few-shot learning 은 주로 모델을 pre-training 시키거나, meta-learning 방법을 이용하는 데, meta-learning 방법에는 network-based, optimization-based 그리고 metric-learning-based 방법으로 구분할 수 있다. Network-based 방법은 네트워크나 별도의 메모리를 사용하여 가중치를 생성하거나, 갱신 또는 예측을 하는 방법이다. Optimization-based 방법은 두 단계의 optimization 절차를 가져 가중치를 초기화 하거나 갱신하는 방법과 같은 학습 과정 자체를 학습한다. 이렇게 해서 최적의 학습 과정으로 few-shot 데이터를 학습을 시작한다 [3]. Metric-learning-based 방법은 기존에 학습했던 지식을 embedding space 에 두어 비슷한 클래스를 가까이 두도록 하는 방법이다 [4, 5, 6, 7].

2.2. Class incremental learning

Class incremental learning 은 여러 태스크에 걸쳐 새로운 클래스가 주어질 때, 기존에 학습했던 클래스에 대한 성능은 유지한 채 새로운 클래스를 잘 배우도록 하는 학습 방법이다. 이때, 기존에 학습했던 클래스를 잊어버리는 현상을 catastrophic forgetting 이라고 한다. Catastrophic forgetting 을 줄이기 위한 방법으로는 generative-replay-based, expansion-based, regularization-based 방법으로 구분할 수 있다. Generative-replay-based 방법은 GAN [8]이나 VAE [9]와 같은 생성 모델을 이용하여 기존에 학습했던 데이터를 재생하는 방법이다 [10]. Expansion-based 방법은 모델의 구조 자체를 늘려 catastrophic forgetting 을 방지하는 방법이다 [11]. Regularization-based 방법은 모델의 가중치를 규제함으로써 기존 태스크에서

학습한 지식을 보존하고자 한다 [12].

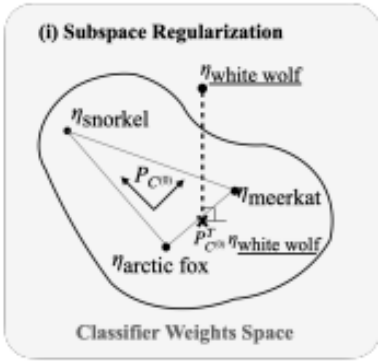
3. 기존 연구 분석

본 프로젝트를 진행하기 앞서 선행 연구인 "Subspace regularizers for few-shot class incremental learning" [13]을 소개하고 분석해보고자 한다.

3.1. 제시된 방법

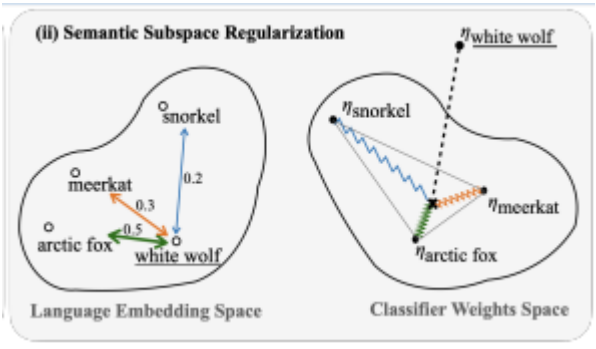
해당 논문에서는 FSCIL 을 다루기 위한 regularization 방법으로 subspace regularization, semantic subspace regularization, linear mapping 을 소개하고 있다.

3.1.1. Subspace regularization



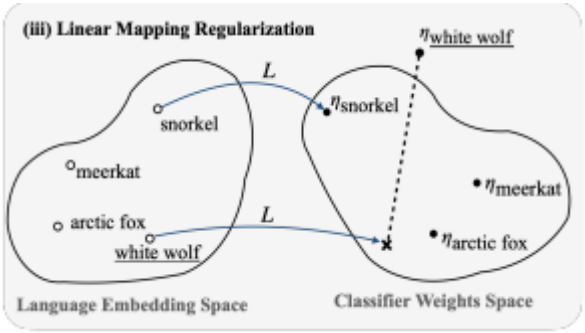
적은 수의 데이터로 이루어진 novel 클래스에 대한 학습은 편향된 정보를 학습할 가능성이 높아진다. 그러나, 의미 있는 정보들은 여러 클래스에 걸쳐 존재하기도 한다. 이에 기반하여, 새로 학습하는 novel classifier weight 를 잘 학습되었다고 볼 수 있는 base classifier weight 로 이루어진 subspace 에 위치하도록 한다면 오버피팅을 줄일 수 있다. 이 방법을 subspace regularization 이라고 명명하고, novel classifier weight 와 해당 novel classifier weight 를 base classifier weight 로 구성된 subspace 에 projection 한 값과의 Euclidean distance 를 계산하여 regularization term 으로 둔다.

3.1.2. Semantic subspace regularization



Subspace regularization 방법에서는 novel classifier weight 를 base classifier weight 로 이루어진 subspace 에 위치시키는 것이 목적이었지, novel classifier weight 를 어디에 위치시켜야 할 지는 고려되지 않았다. 예를 들어, novel 클래스 중 "white wolf"는 base 클래스 중 "snorkel" 보다는 "arctic fox"에 시각적으로 동시에 의미론적으로 더 유사하다. 이러한 부분 또한 학습시에 고려되면 효과적일 것이다. 이를 위해 language embedding 을 사용한다. Language embedding 을 함께 사용한 semantic subspace regularization 에서는 novel classifier 가 subspace 상에서 보다 유사한 base 클래스의 weight 에 가까이 위치하도록 규제한다.

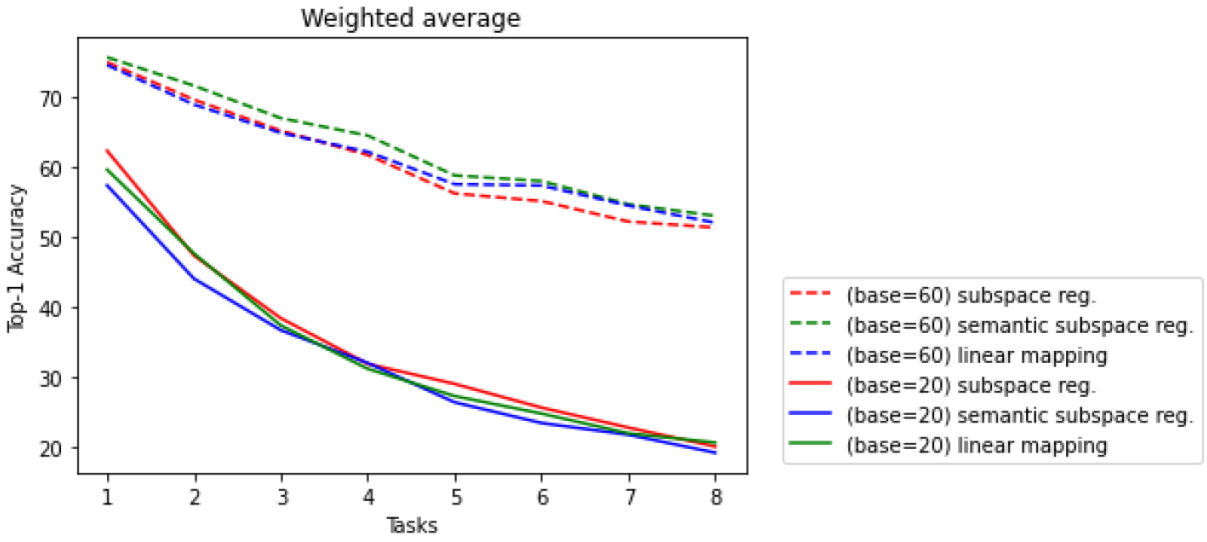
3.1.3. Linear mapping



Zero-shot learning 을 다룬 여러 연구에서는 language embedding 을 바로 classifier weight 에 mapping 하는 방식이 효과가 있음을 보여 준다. 이에 기반하여 해당 논문에서는 language embedding 에서 classifier weight 로의 linear mapping function 을 학습시켜 novel classifier weight 를 해당 클래스의 language embedding 을 입력으로 하는 linear mapping function 의 출력에 가까이 학습하도록 규제한다. 이를 linear mapping 이라고 명명한다.

3.2. 기존 방법의 문제점

앞서 소개된 subspace regularization, semantic subspace regularization 그리고 linear mapping 은 충분한 수의 base 클래스에서 학습된 weight 에 의존한다. 본 연구에서는 이러한 상황에서 만일 base 클래스의 수가 적어질 경우 심각한 성능 저하를 야기할 것이라고 가정했다. 이를 확인해보기 위해 기존 연구에서 base 클래스의 수를 60 으로 설정했던 것과는 달리 base 클래스의 수가 20 일 때 성능을 측정해보았다. 결과는 아래와 같다.



실험 결과, base 클래스의 수가 적어지게 되면 기존 연구에서 소개된 방법이 큰 성과를 거두지 못한 것을 알 수 있었다. 본 연구에서는 이러한 현상에 대한 원인으로 base 클래스의 수가 적어졌기 때문에 신뢰할 만한 subspace 를 형성하지 못했고, language embedding 을 활용함에 있어서도 base 클래스의 수가 적어지면 큰 효과를 거두지 못했기 때문이라고 생각했다.

이에 4 장에서는 이러한 상황에서 성능향상을 위해 시도해 보았던 여러 방법들을 소개한다.

4. 시도해 본 방법

4.1. Reusing novel weights

[13]에서는 base 클래스를 활용한 subspace 을 사용하여 새로 학습될 가중치를 규제하는 방법들을 제시했다. 그러나 이러한 방법은 subspace 가 어떻게 형성되는지에 따라 성능이 크게 변화하게 된다. 따라서 이러한 문제점을 해결하기 위해 각 새롭게 학습한 클래스 또한 subspace 형성하는데 활용할 수 있도록 구현하여 성능을 이끌어 낸다.

Subspace regularization 의 경우 base classifier weight 를 이용하여 subspace 형성 후 새로 학습될 novel classifier weight 를 base classifier weight 로 형성된 subspace 에 projection 시켜준 것을 subspace target 라고 정의한다.

$$m_c = P_{C^{(0)}}^{\mathrm{T}} \eta_c$$

식에서 보면 알 수 있듯이 subspace target 을 구할 때 base 클래스인 $C^{(0)}$만을 사용한다. 이 식에서 기존의 novel 클래스를 재사용 할 수 있도록 다음과 같이 재정의한다.

$$m_c = P_{C^{(<t)}}^{\mathrm{T}} \eta_c$$

Semantic subspace regularization 의 경우 base class embedding 과의 cosine similarity 에 softmax 를 적용한 값을 semantic target 이라고 정의한다.

$$l_c = \sum_{j \in C^{(0)}} \frac{\exp(e_j \cdot e_c/\tau)}{\sum_{j \in C^{(0)}} \exp(e_j \cdot e_c/\tau)} \eta_j$$

semantic target 을 구할 때 base 클래스인 $C^{(0)}$만을 사용하므로 이 시점에서 기존의 novel 클래스를 재사용 할 수 있도록 다음과 같이 재정의한다.

$$l'_c = \sum_{j \in C^{(<t)}} \frac{\exp(e_j \cdot e_c/\tau)}{\sum_{j \in C^{(<t)}} \exp(e_j \cdot e_c/\tau)} \eta_j$$

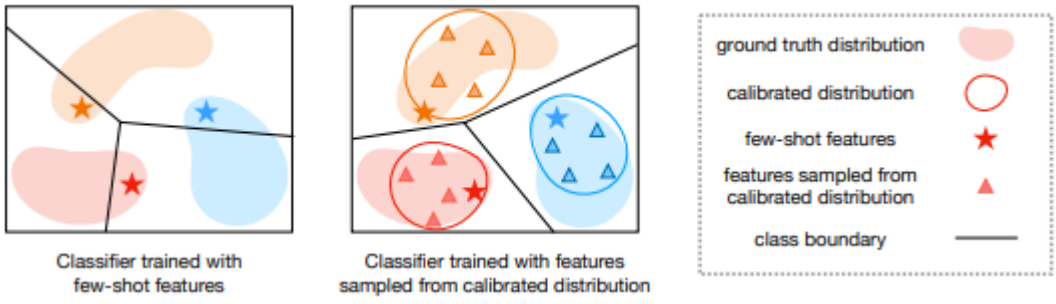
마지막으로 linear mapping 의 경우 base 클래스의 embedding 을 입력으로 하여 클래스의 가중치로 mapping 되도록 하는 선형함수를 학습시켜 최적의 linear function 을 L^*로 정의한다.

$$L^* = \min_L \sum_{j \in C^{(0)}} \|\eta_j - L(e_j)\|^2$$

L^*을 구할 때 base 클래스인 $C^{(0)}$만을 사용하므로 novel 클래스를 재사용 하기 위해서 다음과 같이 재정의한다.

$$L^* = \min_L \sum_{j \in C^{(<t)}} \|\eta_j - L(e_j)\|^2$$

4.2. Distribution calibration



적은 수의 데이터는 biased distribution 을 나타내고, 실제 distribution 을 반영하기는 어렵다. [14]에서는 이를 해결하기 위해 distribution calibration 방법을 제안한다. Base 클래스의 feature 로부터 mean 과 covariance 를 구해준다. 새로운 novel 클래스를 학습할 때, 해당 이미지의 feature 와 가장 유사한 base 클래스를 k개 선정한다. 이때, 유사도의 측정은 novel 클래스의 이미지로부터 얻을 수 있는 feature 와 base 클래스의 feature 에서 mean 값과의 Euclidean distance 이다. 여기서 구해진 k개의 base 클래스를 활용하여 distribution 을 calibrate 한다. 최종적으로 구해진 distribution 에서 feature 들을 sampling 하여 학습에 활용한다.

위 [14]의 방법이 적은 수의 데이터만을 활용하여 학습을 진행하는 FSCIL 환경에서 효과가 있을 것이라 기대하고 실험을 진행했다. 실험 결과는 5 장에서 확인할 수 있다.

4.3. Dark experience replay

Dark experience replay (DER) [15]는 replay-based lifelong learning method 의 일종이다. 해당 논문에서는 이전 태스크의 input image, class label 그리고 output logits 을 메모리에 저장한다. 논문에서는 DER 과 DER++를 제안한다. DER 은 메모리에 저장된 input image 와 output logits 를 사용하여 새로운 태스크를 학습하면서도 메모리의 input image 에 대해 기존의 output logits 과 유사한 출력을 내도록 하는 일종의 knowledge distillation [16] 기법이다. DER++은 DER 의 개선된 버전으로, 새로운 태스크를 학습할 때, 메모리에 저장된 input image 와 class label 을 활용하여 주어진 입력을 계속 잘 분류하도록 규제한다.

그러나 이러한 방법은 이미 [13]에서 적용이 되었다. [13]에서는 3 장에서 소개된 regularization 방법 외에도 여러 규제항을 적용했는데, 예를 들어 아래 수식 같은 경우는 기존에 학습된 classifier 가 너무 바뀌지 않도록 규제한다.

$$R_{old}^{(t)}(\eta) = \sum_{t'<t} \sum_{c \in C^{(t')}} \left\| \eta_c^{t'} - \eta_c \right\|^2$$

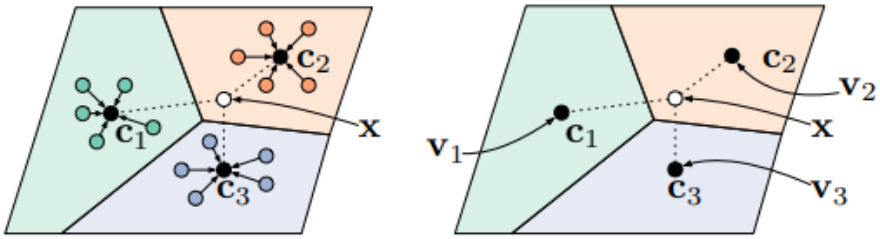
그리고 DER 에서는 아래 항을 통해 이전 태스크의 logit 을 유지하도록 규제한다.

$$\alpha \mathbb{E}_{(x,z)\sim\mathcal{M}} [\| z - h_\theta(x) \|]_2^2$$

[13]의 feature extractor 는 base 클래스로 우선 학습이 되면 이후에는 고정된다. 그렇게 되면 DER 에서 사용된 위 수식은 입력이 같기 때문에 결국 classifier weight 에 따라 값이 변화하게 되고 결국엔 이미 [13]에서 사용된 수식과 같아진다. 또한 DER++에서는 이전 태스크의 input image 와 label 을 replay 하는데 이도 이미 [13]에서 적용이 된 방법이다. 따라서 DER 및 DER++이 본 연구에서는 성능 향상에 도움이 되지 않는다고 판단했다.

4.4. Prototype loss

Prototypical networks [5]는 각 클래스와 prototype representation 사이의 거리를 계산하여 classification 을 수행할 수 있는 metric space 를 학습한다. 본 연구에서는 prototypical networks 를 구현할 때 필요한 class prototype 에 초점을 두었다.



(a) Few-shot (b) Zero-shot

Class prototype 이란 각 클래스에 해당되는 데이터들의 값을 모두 합하여 평균값을 구한 지점을 의미하며 c1, c2, c3 가 여기에 해당된다. [13]의 과정 중 subspace target 을 base 클래스의 가중치로 형성된 subspace 에 정사영을 할 때 class prototype 도 같이 정사영을 시킨 후 두 지점 간 Euclidean distance 를 계산하여 regularization term 으로 활용한다.

5. 실험

5.1. 문제 정의

데이터셋의 전체 클래스는 base 클래스와 novel 클래스로 나눌 수 있다. 태스크가 $T_0 T_1, T_2, \ldots,$ 의 순서로 주어진다. 0 번째 태스크 T_0 에는 base 클래스로 구성되어 있고, 해당 태스크의 데이터를 이용하여 모델을 pre-train 한다. 각 태스크 T_t $(t >= 1)$에서는, 주어진 데이터를 학습 데이터에 해당하는 support set S^t 와 query set Q^t로나눌 수 있다. Support set 에서는 novel 클래스 중 N 개의 클래스가 각각 K 개의 데이터를 가지고 있는데, 이를 N-way, K-shot 환경이라 정의한다. t번째 태스크의 support set 에 대한 학습이 완료되면 0 번째 태스크의 클래스부터 t 번째 태스크의 클래스 모두에 걸친 query set $Q^{<=t}$으로 학습된 모델을 평가한다.

5.2. 학습 설정

모델을 테스트하기 위해 miniImageNet [7] 데이터셋을 사용하였다. miniImageNet 은 기존 ImageNet 의 축소버전으로 100 개의 클래스당 600 개의 이미지로 이루어져 있다. 본 연구에서는 보다 어려운 학습 환경을 설정하기 위해 base classes 의 수를 20 으로, 각 novel task 를 10-way, 5-shot 으로 구성하여 총 8 번의 novel task 로 구성했다. 실험에 사용된 backbone 은 ResNet18 [17]이며 base classes 에 대해서 학습을 마친 이후 feature extractor 를 고정시켰다. 이외에도 많은 설정은 [13]와 일치한다.

5.3. 실험 결과

5.3.1. Reusing novel weights

Task Methods (+M)	Reuse novel	1	2	8	Average
Subspace Reg.	X	**62.27**	47.30	18.98	34.14
	O	**62.27**	47.45	19.62	34.25
Semantic Subspace Reg.	X	57.33	44.00	19.14	32.69
	O	57.33	44.50	**21.56**	33.58
Linear Mapping	X	59.87	**47.60**	18.62	33.26
	O	59.87	46.80	21.12	**34.33**

위 표는 기존에 학습된 novel classifier weight 를 새로운 novel 클래스를 학습할 때, 성능의 차이를 나타낸 표이다. Novel classifier weight 를 사용하지 않았을 때의 실험 환경은 [13]에서와 같다. 실험 결과, novel classifier weight 를 재사용하여 subspace 를 재정의 한 경우가 그렇지 않은 경우보다 모든 regularization 방법에서 효과가 있음을 확인할 수 있었다.

5.3.2. Distribution calibration

Task Methods (+M)	Reuse novel	DC (k=3)	1	2	8	Average
Subspace Reg.	X	X	**62.27**	47.30	18.98	34.14
		O	62.53	**48.30**	19.78	34.09
Semantic Subspace Reg.		X	57.33	44.00	19.14	32.69
		O	57.67	44.15	19.72	32.37
Linear Mapping		X	59.87	47.60	18.62	33.26
		O	58.67	47.50	19.96	33.32
Subspace Reg.	O	X	**62.27**	47.45	19.62	34.25
		O	62.53	47.55	19.86	33.87
Semantic Subspace Reg.		X	57.33	44.50	**21.56**	33.58
		O	57.67	44.50	20.94	33.14
Linear Mapping		X	59.87	46.80	21.12	**34.33**
		O	58.67	47.40	20.42	33.53

위 표는 4.1 의 방법과 4.2 절에서 소개된 distribution calibration (DC)의 적용 여부에 따른 성능 차이를 나타낸 것이다. 실험 결과, 4.2 절의 기대와는 반대로 distribution calibration 을

적용시켰을 때 성능이 다소 하락하는 현상을 확인할 수 있었다.

5.3.3. Protype loss

Task Methods (+M)	Prototype loss	1	2	3
Subspace Reg.	X	**62.27**	**47.45**	**37.48**
	O	61.49	47.23	36.89

위 표는 4.4 에서 소개된 prototype loss 를 적용시킨 결과이다. 새로운 방법을 적용했음에도 불구하고 성능 하락을 보였다.

6. 결론 및 한계점

본 연구에서는 few-shot class incremental learning 을 다루었다. Few-shot class incremental learning 에서 최신 연구를 분석하고 해당 연구에서 소개된 방법이 base classes 의 수가 적은 상황에서 제대로 작동하지 않는다는 점을 발견하였다. 이를 해결하기 위해 학습된 novel classes 의 weight 를 재사용하는 새로운 방법을 제안하거나, few-shot learning 과 lifelong learning 분야에서 연구되었던 여러 방법들을 적용해보았다. 기존의 방법들은 큰 성과를 거두지 못하였으나, novel classes 의 weight 를 재사용하는 방법은 약간의 성능 향상을 이루었다.

본 연구에는 한계점 또한 존재한다. 우선 선행 연구인 subspace regularization 을 개선시키기 위한 알고리즘을 새로 고안한 것이기 때문에 다른 few-shot class incremental learning method 에 효과가 있음을 장담할 순 없다. 또한, 성능 향상의 폭이 아주 작다는 점이 아쉬움으로 남았다.

그럼에도 불구하고, 본 연구에서 제시된 방법은 매우 간단하지만 효과적이므로, 향후 subspace 를 활용한 few-shot class incremental learning method 에 관한 연구에 도움이 될 것이라고 생각한다.

7. 참고 문헌

[1] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). IEEE.

[2] Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., ... & Schmidt, L. (2022). Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *arXiv preprint arXiv:2203.05482*.

[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In ICML, 2017.

[4] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In ICMLW, 2015

[5] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In NIPS, 2017.

[6] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In CVPR, 2018.

[7] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In NIPS, 2016.

[8] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

[9] Diederik P. Kingma, Max Welling, Auto-encoding Variational Bayes, In IEEE Conference on Learning Representations, 2014.

[10] Chensen Wu, Luis Herranz, Xialei Liu, Yaxing Wang, Joost van de Weijer, Bogdan Raducanu, Memory replay GANs: learning to generate images from new categories without forgetting. In NeurIPS 2018.

[11] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks. arXiv preprint arXiv:1606.04671, 2016.

[12] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming Catastrophic Forgetting in Neural Networks. Proceedings of the National Academy of Sciences, 2017.

[13] Afra Feyza Akyürek, Ekin Akyürek, Derry Wijaya, Jacob Andreas, Subspace regularizers for few-shot class incremental learning, In ICLR 2022.

[14] Shuo Yang, Lu Liu, Min Xu, Free Lunch for Few-shot Learning: Distribution Calibration, In ICLR 2022.

[15] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, Simone Calderara, Dark Experience for General Continual Learning: a Strong, Simple Baseline, In NeurIPS 2020.

[16] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, Distilling the Knowledge in a Neural Network,

In NeurIPS 2014.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, In CVPR 2016.