

# Continual Learning 방법론 분석 및 개선

이름 홍건화

지도교수 황원준

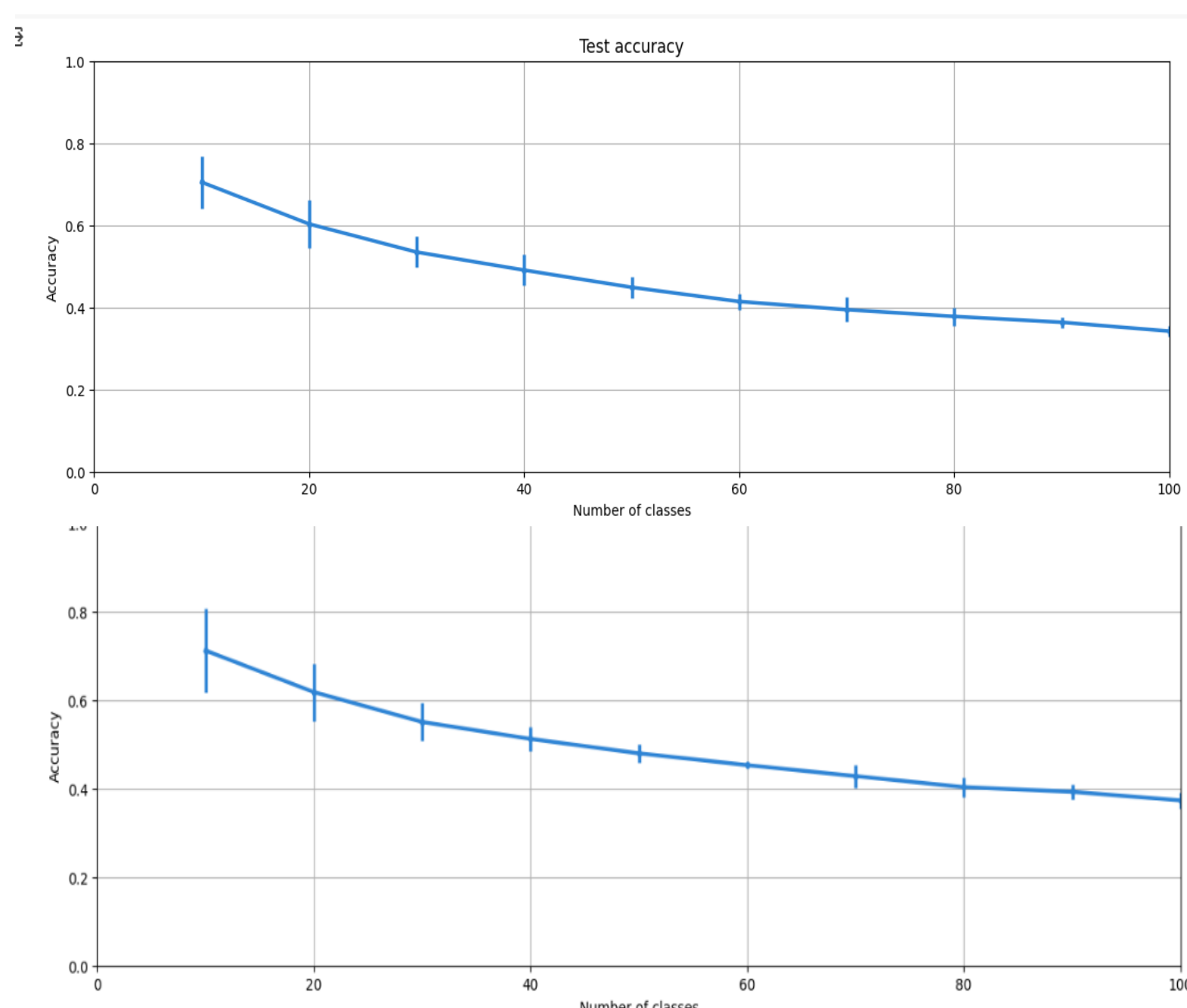
## 연구배경

과거의 시는 데이터를 학습할 때 일괄적으로 주입하고 변화 없이 사용하는 경향이 있다. 그러나 지속적으로 증가하고 변화하는 데이터를 효과적으로 다루기 위해서는 데이터가 들어올 때마다 추가적인 학습이 가능한 시스템이 필요하다. 이를 위해 DataStream과 Model의 변화를 효과적으로 다루기 위한 방법들이 개발되었다. Continual Learning은 모델이 새로운 데이터셋을 지속적으로 학습하면서

Catastrophic Forgetting과 Semantic Drift를 방지하는 목적을 가지고 있다. Catastrophic Forgetting은 모델이 새로운 정보를 학습하면서 이전에 학습한 정보를 완전히 잊어버리는 현상이며, Semantic Drift는 모델이 학습한 의미나 패턴이 시간이 지남에 따라 변하는 현상이다.

Continual Learning을 위한 다양한 방법론이 제안되어왔고, 각 방법들의 장단점을 분석하여 효율적인 개선 방안을 탐색한다.

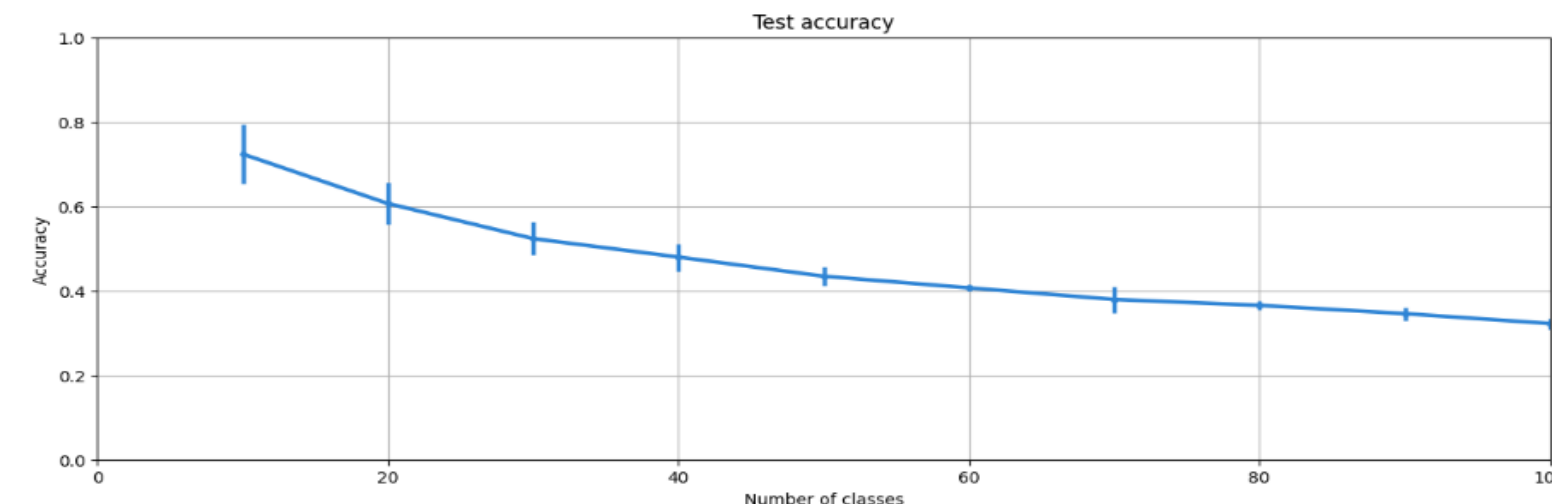
## 결과 및 분석



20개의 exemplar를 학습하여 만들었던 결과와 20\*5를 통한 학습 결과 대략 1~1.5% 정도의 성능 향상을 확인할 수 있었다. 단순 복사를 통한 데이터 추가를 통하더라도 이전 Task에 대한 정보를 전달하는 것에는 큰 문제가 없음을 확인

할 수 있었다. 하지만 auto augmentation으로 진행하여 80개를 추가한 결과는 20개 와 비교해도 미세하게 하락했다.

Rand\_augmentation으로 실행한 경우는 미세하게 정확도가 향상되었다. 이를 토대로 추측 할 수 있는 것은 data augmentation은 변조를 통해 의외의 data를 만들어서 각 Task의 dataset을 보충하여 학습을 풍부하게 할 수 는 있으나 exemplar 에 응용하려는 경우 해당 Task의 대표성에서 떨어지는 값들이 될 수도 있어 그 효과가 떨어질 수도 있고 augmentation의 방식에 따라 그 대표성을 유지하여 효과가 늘어날 수도 있다고 생각할 수 있다.



## 향후 연구 방향

다양한 Data Augmentation 기법을 사용하여 이전 Data의 정도를 최대한 유지할 수 있는 간단한 augmentation을 사용하여도 효과가 감소하는지 확인하고 추가적인 다양한 augmentation 실험을 수행한다, 단순 복사를 통한 데이터 증식의 수치를 조정하여 어느 정도까지 효과가 증진하는지 확인한다.

이후, Structure Method를 단순히 추가하는 것이 아니라 특정 기준을 정하여 Column을 추가하면서 동시에 exemplar를 처리하는 Hybrid 방식을 실험하여 효과를 평가한다.

## 연구진행과정

Elastic Weight Consolidation (EWC)은 정규화 방법 중 하나로 간단한 Loss 계산식의 변경만으로 구현할 수 있지만 모델이 이전에 학습한 작업에 대한 학습을 보존하는 데 중점을 두기에 모델이 이전 작업에서 학습한 가중치를 보호하기 위해 새로운 작업에 대한 학습을 어렵게 만든다. 따라서 지속적으로 작업이 추가되는 환경에서는 새로운 작업의 학습이 방해되고, 결과적으로 해당 작업의 정확도가 떨어질 수 있다. 새로운 학습에 대한 중요성을 높이는 등의 과정이 필요하다.

Rehearsal Method 중 하나인 iCaRL은 모델이 이전에 학습한 작업에서의 Data를 일부 보존하고 새로운 작업에 대한 학습을 진행하는 방법으로, 저장 메모리의 크기를 늘릴 경우 유의미한 성능 향상을 얻을 수 있다. 그러나 Task가 지속적으로 늘어날 경우 할당하는 메모리의 용량이 계속해서 증가하고 메모리 한계를 초과할 수 있다. 이를 막기 위해 Task의 exemplar 할당량을 줄이게 되어 전반적인 성능 하락이 발생한다. 메모리를 효과적으로 사용하기 위한 적절한 개선안이 필요하다.

Generator model을 별도로 학습하여 exemplar를 생성하는 DGR(Dynamic Generative Replay)방식은 메모리 제한에 구애 받지 않고 다양한 클래스에 대한 exemplar를 사용할 수 있고 Task가 지속적으로 들어오더라도 메모리에 문제가 발생하지 않지만, Generator Model에서 발생하는 forgetting을 처리할 필요가 있으며 학습 모델이 추가되기 때문에 학습시간이 크게 증가한다.

Progressive Neural Network(PNN)은 새로운 Task가 추가될 때마다 모델에 새로운 Column을 추가하는 structure method로. 이전 Column의 값을 고정하여 이전 학습의 가중치를 저장한다. 이 방식으로 Task가 계속해서 추가되더라도 모델의 크기만 반복적으로 늘리면 해결이 가능하다. 이전 Task의 보존이 확실 하지만 이전 column의 접근이 제한되기에 Transfer learning의 효과가 약해지고 Model을 효율적으로 쓰지 않고 단순 추가만 진행하기에 model이 비효율적으로 커지게 된다. Column의 추가를 결정하는 기준을 개선하는 것이 필요하다.

DGR과 iCaRL에서 exemplar를 활용한 학습이 Continual Learning에 효과적임을 확인했고, 특히 DGR에서 생성된 exemplar가 실제로 선정한 exemplar와 유사한 효과를 보이기에, exemplar에 인위적으로 접근 하는 방식 또한 Task의 대표성을 유지한다면 continual learning으로 이전 Task의 정보를 유지하는 것에는 효과적임을 확인할 수 있었다. 그러나 생성형 모델 학습에 큰 Cost가 드는 DGR을 사용하는 것보다는 iCaRL과 같이 실제 데이터에서 exemplar를 선정하고 단순한 복사 또는 Data Augmentation을 통해 exemplar를 늘리는 방법을 통해 성능을 높이고 저장 memory를 효율적으로 사용할 것이라는 결론을 도출했다.

실험에서는 ResNet32 모델과 CIFAR-100 데이터셋을 10개의 클래스로 분할한 10개의 Task를 iCaRL 코드를 통해 실행하고, 그 Test 결과를 분석하였다. Exemplar의 선정은 iCaRL에서 Herding 규칙에 맞춰서 선정한 20개를 단순 복사를 진행시키거나 다양한 Data Augmentation을 통해 80개를 추가해 총 100개로 만들어서 진행하고 data augmentation의 방식에 따라 변화하는지 또한 확인한다.