

데이터 유사도를 이용한 지속적 학습방법

Continual Learning using Data Similarity

박 성 현*, 강 석 훈*

Seong-Hyeon Park*, Seok-Hoon Kang*

Abstract

In Continuous Learning environment, we identify that the Catastrophic Forgetting phenomenon, which forgets the information of previously learned data, occurs easily between data having different domains.

To control this phenomenon, we introduce how to measure the relationship between previously learned data and newly learned data through the distribution of the neural network's output, and how to use these measurements to mitigate the Catastrophic Forgetting phenomenon. MNIST and EMNIST data were used for evaluation, and experiments showed an average 22.37% improvement in accuracy for previous data.

요 약

Continuous Learning 환경에서 인공 신경망의 학습이 진행됨에 따라 이전에 학습했던 데이터의 정보를 잊는 Catastrophic Forgetting 현상이 있다. 서로 다른 Domain을 갖는 데이터 사이에서 쉽게 발생한다. 이 현상을 제어하기 위해 신경망의 출력 분포를 통해 이전에 학습된 데이터와 새로 학습할 데이터들의 관계를 측정하는 방법과 이 측정값을 사용하여 Catastrophic Forgetting 현상을 완화하는 방법을 제시한다. 평가를 위해 MNIST, EMNIST 데이터를 사용하였고 실험 결과, 이전 데이터에 대한 정확도가 평균적으로 약 22.37% 향상되었다.

Key words : Continuous Learning, Catastrophic Forgetting, Neural Network, Deep Learning, Regularization

1. 서론

CL(Continuous Learning)은 새로운 환경에 적응하고 복잡한 기술과 지식을 습득할 수 있는 인공지능 에이전트를 생성하기 위한 중요한 단계이다[1]. CL 관련 기술의 특징 중 하나는 최신 데이터를 사용하여 모델을 계속 업데이트해야 한다는 것이다. 그러나 이전 학습 데이터를 메모리에 계속 보존하며 학습하는 것은, 시스템의 구현 관점에서 힘든

것이다. 또한 신경망이 새로운 데이터에 대해서만 학습하여 가중치를 업데이트 하면, 치명적 망각(Catastrophic Forgetting) 현상이 발생하여, 이전 데이터에 맞춰진 가중치 분포를 재구성하게 되고 이전 데이터에 대한 정보를 잊게 된다[2]. 이러한 현상을 완화하기 위해 여러 가지 전략이 제안되었다. 그러나 치명적 망각은 여전히 효과적인 CL 구현을 향한 주요 장애물이다. 특히 새로운 데이터가 이전 데이터의 분포와 상반된 분포를 보인다면 이

* Dept. of Embedded Systems Engineering, Incheon National University

★ Corresponding author

E-mail : hana@inu.ac.kr, Tel : +82-32-835-8422

※ Acknowledgment

This work was supported by Incheon National University Research Grant in 2020 (2020-0121)

Manuscript received Jun. 1, 2020; revised June. 14, 2020; accepted June. 17, 2020.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

러한 현상은 더 심해지게 된다[3, 4]. 망각현상을 제어하기 위해서는 이전 데이터와 새로운 데이터를 같이 고려하는 방법이 필요하다. 본 논문에서는 새로운 데이터를 학습할 때, 기존에 학습된 데이터와의 유사도가 CL 관점에서 의미있는 관련성이 있음을 실험적으로 제시하고, 치명적 망각을 완화시키는 주요방법 중 하나인 EWC(Elastic Weight Consolidation) 방법에 이 결과를 적용하여, 실험적 환경에서 효과적으로 망각현상을 완화시킬 수 있음을 보인다. 본 논문의 결과는 가변적인 비용함수의 기반 데이터가 되어 효율적인 함수 적용에 의한 망각현상 개선에 이용될 수 있다.

II. 본론

1. 관련 연구

CL 시나리오는 크게 2가지로 나누어 생각할 수 있다[5, 6]. 첫번째는 대부분의 실험에서 사용하는 Multi-Task(MT) 시나리오다. 이 방법은 독립적인 Task들을 점진적으로 학습하며, 이전에 배웠던 Task들을 잊지 않아야 한다. Task간 Class의 중복은 없으며 정확도는 각 Task가 별도로 계산된다. 즉, 이 방법은 이전의 과제를 해결하는 방법을 잊지 않고 분리된 Task에 대해 단일 모델을 훈련하는 타당성을 연구하는 데 적합하지만 증분 문제를 해결하는 데는 적합하지 않다. 두번째는 Single-Incremental-Task(SIT) 시나리오다. 이 방법은 점진적인 단일 Task를 고려한다. SIT의 예로 클래스 증분 학습(NC)이 있다. 여기서 MT 방법과 같이 새 클래스를 차례로 추가하지만, 정확도를 계산할 때 지금까지 학습 한 모든 클래스를 구별해야 한다. 이러한 방법은 인간이 새롭게 학습한 객체를 이전에 배웠던 객체들과 구별을 할 수 있듯이 실제 세계에서 자연스러운 방식이다. 일반적으로 SIT 방법이 MT 방법보다 어렵다. EWC에서 변형된 MNIST 데이터로 실험한 방법은 SIT 시나리오의 NI 방법으로 생각할 수 있다.

SIT 시나리오는 구체적으로 3가지 학습 방법으로 나눌 수 있다.

- New Instance(NI) : Class는 고정, 모든 Class의 Instance를 점진적으로 학습
- New Classes(NC) : 새로운 Class를 점진적으로

학습

- New Instances and Classes(NIC) : 새로운 Class와 이전에 학습된 Class의 Instance를 점진적으로 학습

EWC는 이전 Task에서 학습된 모델의 매개변수의 중요도를 FIM(Fisher information matrix)으로 정의하고, 이를 파라미터당 가중치 감소값으로 활용하여, 중요도에 따라 매개변수가 다음 Task 학습에서 많이 변하지 못하도록 정규화시키는 방법이다.

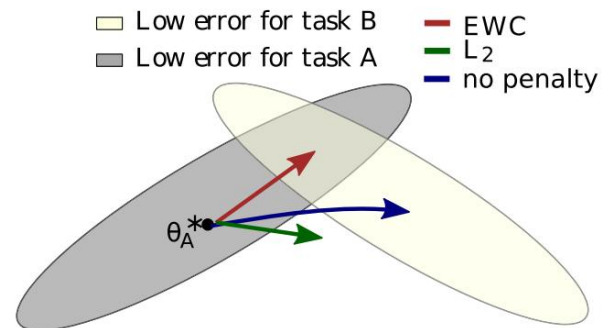


Fig. 1. Basic EWC Idea.

그림 1. EWC의 기본

그림 1에서 회색의 타원형 부분은 Task A에 최적화된 매개변수(가중치) 영역이다. 현재 모델이 Task A를 학습하여 회색 영역에 있고, 이 상태에서 Task A를 기억한다는 목적 없이 Task B를 학습한다면 매개변수의 변화 궤적은 파란색 화살표대로 이동할 것이다. 이러한 매개변수의 변화를 억제하고자 매개변수가 잘 변화하지 못하도록 모든 매개변수에 제약을 설정한다면 초록색 화살표의 궤적대로 움직여 Task A와 Task B에 대한 최적 매개변수값을 얻을 수 없을 것이다. 만약 Task A에 연관성이 높은 매개변수에만 제약을 설정한다면 빨간색 화살표의 궤적을 따라 Task A와 Task B 모두 잘 수행하는 매개변수 영역을 찾을 수 있을 것이다. 이를 위한 비용함수는 아래와 같은 식으로 정의된다.

$$L = L_B + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta_{A,i}^*)^2 \tag{1}$$

여기서 L_B 는 Task B(현재 학습 데이터)에 대한 비용함수, i 는 각 매개변수의 라벨 값, F_i 는 매개변

수 θ_i 의 이전 Task(Task A)에 대한 상관도, λ 는 이전 Task의 중요도 값으로, 정규화 강도를 나타내며, $\theta_{A,i}^*$ 는 이전 Task(Task A)에 맞춰진 i 번째 매개변수를 나타낸다. 즉, 어떠한 매개변수가 이전 Task에 대한 상관도가 높다면 이 매개변수의 변화에 높은 제약이 걸리게 되므로 변화를 억제할 수 있다. F_i 를 계산할 때 피셔 정보 행렬(Fisher information matrix)의 대각 항을 사용한다.

2. 본 논문의 방법

NI 방법을 사용한 학습에서 EWC와 같은 다양한 정규화 기법들은 우수한 성능을 보인다[3]. 하지만 IDL(Incremental Domain Learning) 환경에서는 낮은 성능을 보인다[4]. IDL 환경은 이전 입력 데이터와 다른 Domain을 가진 데이터가 Task마다 학습 데이터로 들어오는 환경이다.

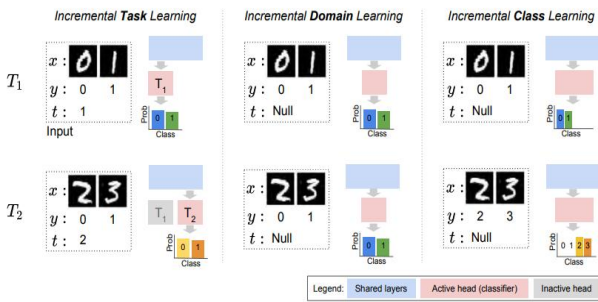


Fig. 2. 3 Learning Scenario for IDL. 그림 2. IDL환경의 3가지 학습 시나리오

[4]의 3가지 학습 시나리오는, 그림 2와 같이 순서대로 MT, SIT/NI, SIT/NC 방법에 해당한다. 실제 환경에서 모델이 CL을 수행할 때 비슷한 형태의 데이터들이 연속으로 들어오는 환경이 아닌, 서로 다른 형태를 가진 데이터가 들어오는 IDL 과 같은 상황은 언제든지 발생할 수 있다.

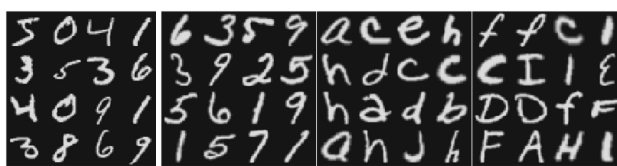


Fig. 3. An Example of MNIST & EMNIST Data. 그림 3. MNIST 와 EMNIST Data의 예

데이터에 따라 망각의 정도를 알아보기 위해, 그림 3과 같은 MNIST[7], EMNIST[8] 데이터를 사용하여 같은 Domain의 데이터가 들어오는 환경과 IDL 환경처럼 다른 Domain의 데이터가 들어오는 환경을 구성하였고, 표 1과 같은 환경으로 실험하였다. 테스트한 결과는 그림 4에 나타낸다.

Table 1. Experiment Scenario using EMNIST Data.

표 1. EMNIST 데이터를 사용한 실험 구성

Task Scenario		A	B
Same Domain	a	MNIST	EMNIST(number)
	b	EMNIST(lowercase)	EMNIST(uppercase)
	c	EMNIST(uppercase)	EMNIST(lowercase)
Different Domain	d	MNIST	EMNIST(lowercase)
	e	MNIST	EMNIST(uppercase)
	f	EMNIST(lowercase)	MNIST
	g	EMNIST(uppercase)	MNIST

EMNIST 데이터는 숫자, 영문자(대문자/소문자) 데이터로 이루어져 있다. 이를 사용하여 같은 Domain을 학습할 경우 MNIST의 숫자 데이터와 EMNIST의 숫자 데이터로 각 Task를 구성하고 EMNIST의 영문자를 대문자, 소문자로 나누어 각 Task를 구성하여 테스트한다. 다른 Domain의 경우 숫자 데이터와 영문자 데이터의 다양한 조합으로 각 Task를 구성하여 테스트한다.

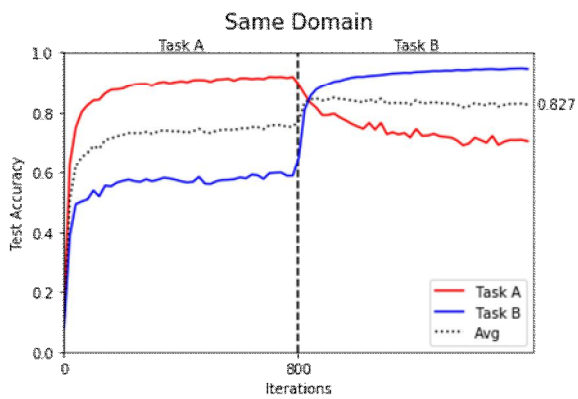
Table 2. Average Acc caused by Domain.

표 2. Domain에 따라 발생하는 망각의 차이

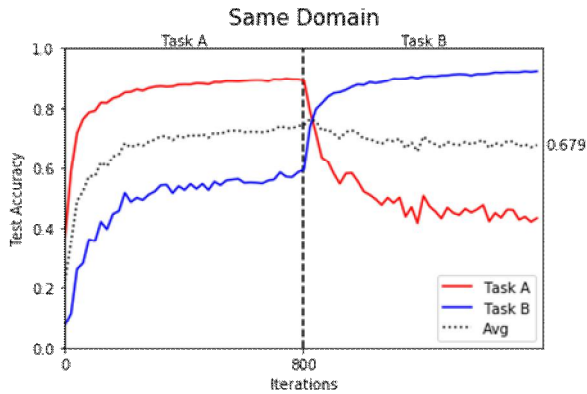
method	Same Domain			Different Domain			
	a	b	c	d	e	f	g
SGD	82.7	67.9	79.2	54.6	53.0	54.7	56.8

그림 4는 SGD(stochastic gradient descent)를 사용하여 학습하는 일반적인 딥러닝 모델 네트워크로 실험한 그래프이다. Task A 실선은 Task A에서 사용한 데이터의 정확도를 나타내고, Task B 실선은 Task B에서 사용한 데이터의 정확도를 나타낸다. 검은색 점선은 전체 Task의 평균이다. 실험결과에 따른 망각의 차이를 표 2에 나타내었다. 표 2에서 같은 Domain을 가진 데이터로 학습할 경우 두 Task의 평균 정확도가 약 68~83%이다. 다

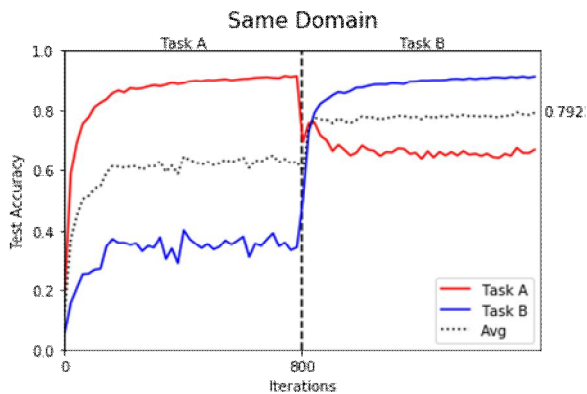
른 Domain을 가진 데이터로 학습한 경우 나타나는 평균 정확도가 약 53~57%인 것에 비해 15%~26% 큰 값을 보인다. 또한, Task A 학습 시 Task B의 정확도가 올라가는 모습을 볼 수 있는데 이는 두 Task 데이터들의 형태가 유사하고 Domain이 같기 때문이다. 이러한 결과를 통해 이전 데이터와 같은 Domain의 데이터를 사용한 경우 망각현상은 비교적 두드러지게 나타나지 않지만, 이전 데이터와 다른 Domain인 데이터가 들어오게 되면 망각현상이 두드러지게 나타나게 된다는 것을 알 수 있다.



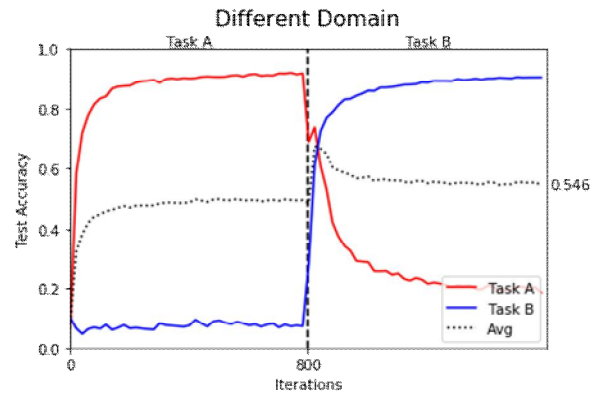
(a) MNIST-EMNIST Number (Same Domain)



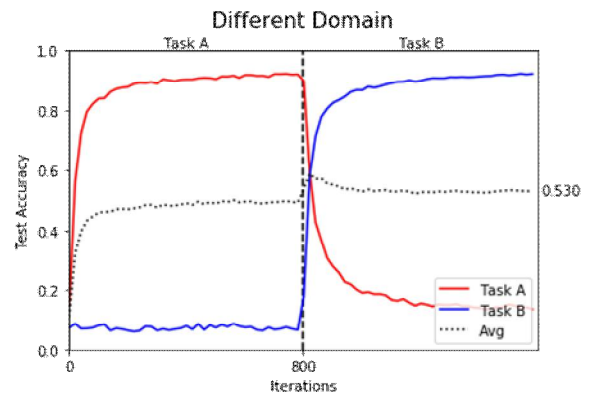
(b) lower case-upper case (Same Domain)



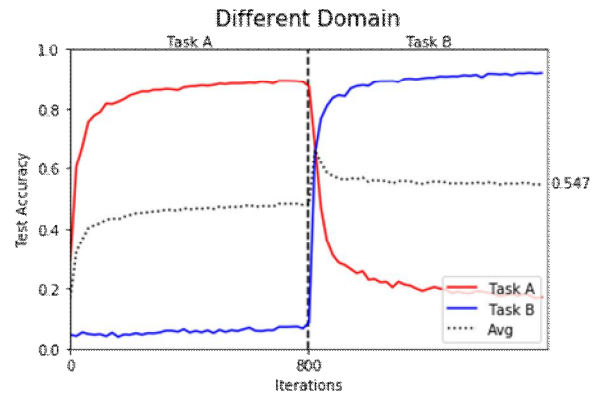
(c) upper case-lower case (Same Domain)



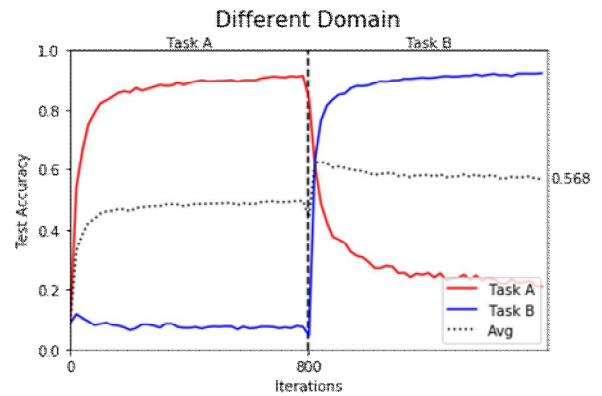
(d) MNIST-lower case (Different Domain)



(e) MNIST-upper case (Different Domain)



(f) lower case-MNIST (Different Domain)



(g) upper case-MNIST (Different Domain)

Fig. 4. Forgetting Graph caused by Domain.

그림 4. Domain 차이에 따라 발생하는 망각 그래프

이 실험을 바탕으로, 다양한 Domain이 입력으로 들어오는 실제 CL 환경에서 신경망이 학습을 잘 하기 위해서는 이전에 학습한 데이터와 새로 학습할 데이터의 특징을 비교하여 유사도를 측정하고, 특징의 차이가 클수록 매개변수의 변화를 억제하는 방법이 해결책이 될 수 있다. 이를 구현하기 위해 각 데이터 사이의 유사도를 측정해야 한다. 단순히 각 Task의 데이터를 직접 비교하는 방법을 생각할 수 있지만, 실제 모델이 학습하는 환경은 CL 환경임을 알고 있어야 한다. CL 환경은 학습이 완료된 이전 Task 데이터에 접근할 수 없는 환경이므로 각 Task의 데이터를 직접 비교할 수 없다. 따라서 이전 Task 데이터에 접근하지 않고 현재 Task 데이터와 특징을 비교할 수 있는 최적적인 방법을 생각해야 한다.

이전 데이터에 직접 접근이 불가능하다면 이전 데이터로 학습된 모델의 출력 분포를 비교하는 방법을 사용하면 유용할 수 있다. 이전에 학습된 데이터와 유사한 형태를 가지는 데이터를 모델에 입력하게 되면 은닉층의 출력 분포 또한 이전 데이터와 유사하게 나타날 것이다. 본 논문에서는 유사도를 측정결과를 적용함으로써 망각이 덜 이루어 지도록 한다.

이를 확인할 실험에 앞서 검증을 위한 Permuted MNIST 데이터를 추가로 사용하였다. 그림 5에 나타난 것과 같이, Permuted MNIST 데이터는 MNIST 이미지의 픽셀 순서를 고정된 패턴으로 재배열하여 새로운 Task의 학습 데이터로 사용한다. 이는 새로운 Task에서 이전의 Task와 형태가 다른 이미지 데이터를 사용한 것과 같은 효과를 낸다.

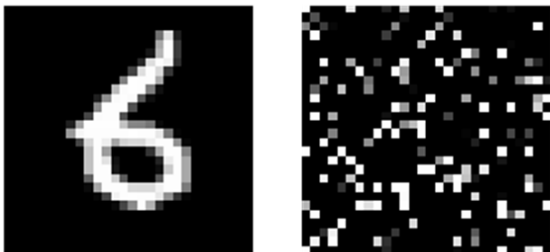


Fig. 5. An Example of Permuted MNIST Data.
그림 5. Permuted MNIST 데이터의 예

이 데이터를 사용하면 이전 Task에서 학습한 데이터의 지식을 얼마나 유지하는지 평가하는 지표로 사용할 수 있다[9, 10].

다음은 이를 확인하기 위해 MNIST로 학습된 모델에 MNIST, EMNIST, Permuted MNIST를 사용하여 출력 분포를 확인하는 실험을 수행한 결과이다.

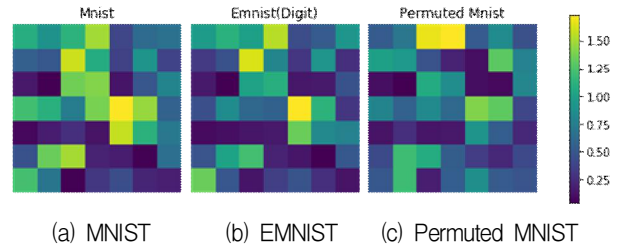


Fig. 6. Output Distribution of Hidden Layer.

그림 6. 각 데이터에 따른 은닉층의 출력 분포 시각화

그림 6은 MNIST 데이터로 학습된 모델에 3가지 데이터들의 표본을 200개씩 뽑아 입력으로 넣은 후, 은닉층의 출력 분포를 평균하여 시각화한 사진이다. (a)는 MNIST 데이터의 출력 분포, (b)는 EMNIST 숫자 데이터의 출력 분포 (c)는 Permuted MNIST 데이터의 출력 분포이다. MNIST 데이터의 출력 분포를 기준으로 2가지 데이터와 상관 계수를 측정했을 경우 MNIST와 유사한 형태를 보이는 EMNIST의 숫자 데이터는 약 0.92의 상관 계수를 가지며, 다른 형태를 보이는 Permuted MNIST는 0.53의 상관 계수를 가진다. 즉, MNIST와 EMNIST의 숫자 데이터의 출력 분포는 강한 연관성을 보이지만 Permuted MNIST는 약한 연관성을 보인다. 출력층의 분포를 이용하면 직접 데이터에 접근하지 않고 각 데이터 사이의 연관성을 비교하는 지표로 사용할 수 있게 된다. 출력 분포를 비교하기 위해 이전 학습 데이터들로 모델이 학습 완료되면 이

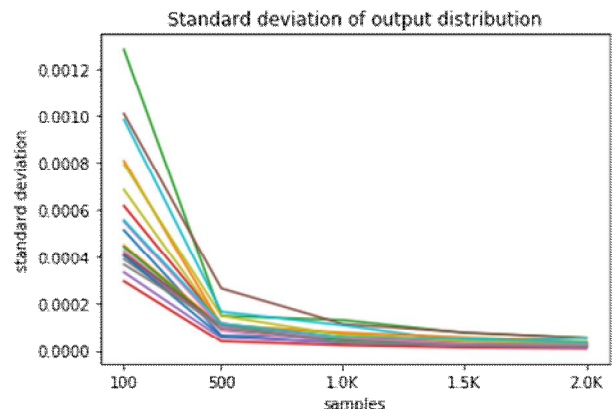


Fig. 7. Standard Deviation graph of the output distribution.
그림 7. 표본 개수에 따른 출력 분포의 표준편차 그래프

전 학습 데이터에서 일정 개수의 표본을 추출하여 출력 분포를 계산하고 저장한다.

이때 사용되는 표본의 개수를 정하기 위해 수행한 실험과 그 결과는 그림 7과 같다.

Table 3. Standard Deviation of the output distribution.

표 3. 표본 개수에 따른 출력 분포의 표준편차 값

Samples	100	500	1000	1500	2000
Standard deviation	0.00059	0.00011	0.00006	0.00004	0.00003

표 3의 학습 데이터, 검증 데이터와 같이 MNIST와 EMNIST 데이터의 다양한 조합으로 실험을 수행하였고, 표본을 뽑아 출력 분포의 상관 계수를 계산하는 과정을 100번씩 반복하여 결괏값의 표준편차를 계산하였다. 그림 9의 그래프를 통해 추출하는 표본의 개수가 많아질수록 표준편차가 줄어드는 모습을 보인다. 추출하는 표본의 개수가 1000개일 경우 모든 조합의 표준편차 평균값은 0.00006 값을 갖고, 이후 평균 표준편차의 변화량이 0.00002 이하로 낮아지므로 표본 추출 개수를 1000개로 설정하여 출력 분포를 계산한다.

이전 학습 데이터에서 출력 분포를 계산하고 새로운 학습 데이터들이 들어오게 되면 학습을 수행하기 전, 같은 방법으로 출력 분포를 계산하여 출력 분포들의 상관 계수를 계산하게 된다. 이를 구현하기 위해 입/출력 계층을 제외한 은닉층의 출력(이 실험에서는 50개의 출력) 분포를 비교한다. 표 4는 다양한 데이터로 모델을 학습한 후 다른 데이터들과 출력 분포의 상관 계수를 측정된 결과이다.

Table 4. Correlation Coefficient of output distribution.

표 4. 각 데이터에 따른 출력 분포의 상관 계수 측정

Test Data \ Train Data	MNIST	EMNIST (N)	EMNIST (L)	EMNIST (U)	MNIST (P)
MNIST	1	0.928	0.914	0.879	0.718
EMNIST (N)	0.916	1	0.859	0.958	0.629
EMNIST (L)	0.870	0.826	1	0.933	0.594
EMNIST (U)	0.892	0.955	0.938	1	0.634
MNIST (P)	0.513	0.610	0.737	0.562	1

표 4의 측정 결과에서, 숫자 데이터 사이의 상관관계와 영문 데이터 사이의 상관관계가 높게 측정되는 것을 알 수 있다. 그러나 숫자 데이터와 영문 데이터 사이의 상관관계 또한 높게 측정이 되었는데 이는 숫자와 영문자의 형태가 유사하므로 발생하는 결과이다. 영문 'i'와 숫자 '1'은 비슷한 형태를 가진다. Permuted MNIST의 경우 다른 데이터와 형태가 유사하지 않으므로 상관 계수가 낮게 나오는 모습이다. 즉, 유사한 형태를 가진 데이터들은 높은 상관 계수를 가지는 경향이 있음을 알 수 있다.

하지만 상관 계수를 바로 유사도로 사용하기에는 무리가 있는데, 숫자 데이터와 영문 데이터 사이의 상관관계 측정 결과와 같이 Domain이 달라도 출력 형태가 비슷한 경우 상관 계수가 높게 나오기 때문이다. 출력 분포의 상관도가 높다면 모델의 최종 출력 값도 유사한 분포를 보일 것이고, 최종 출력 값의 오류가 많다면 오류를 계산하여 역전파 하는 과정에서 기존 데이터에 맞춘 매개변수 값들이 크게 변화될 위험이 있다. 즉, 상관 계수가 높다면 오류 값이 이전 데이터에 미치는 영향이 높고, 상관 계수가 낮다면 오류 값이 이전 데이터에 미치는 영향이 낮다는 것이다. 따라서 상관 계수와 오류 값을 알 수 있으면 데이터가 치명적 망각을 일으키는 정도를 파악할 수 있다. 상관 계수가 -1~1 사이의 값을 가지므로, 본 논문에서는 0~1 범위 값을 가지는 정확도를 오차값으로 대체하여 사용한다.

정확도 계산에서 사용할 표본의 개수를 정하기 위해 이전 실험과 같은 실험을 수행하였고 그 결과는 그림 8과 표 5와 같다.

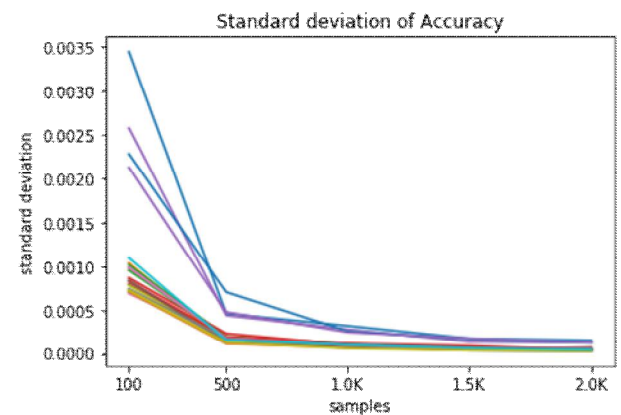


Fig. 8. Standard Deviation graph of Accuracy. 그림 8. 표본 개수에 따른 정확도의 표준편차

Table 5. Standard Deviation of Accuracy.

표 5. 표본 개수에 따른 정확도의 표준편차

Samples	100	500	1000	1500	2000
Standard deviation	0.00121	0.00024	0.00013	0.00008	0.00006

이전 실험과 마찬가지로 그림 8의 그래프를 통해 추출하는 표본의 개수가 많아질수록 표준편차가 줄어드는 것을 알 수 있다. 추출하는 표본의 개수가 1000개일 경우 모든 조합의 표준편차 평균값은 0.00013 값을 갖고, 이후 평균 표준편차의 변화량이 0.00007 이하로 낮아지므로 표본 추출 개수를 1000개로 설정하여 출력 분포를 계산한다. 표 8은 각 데이터로 측정된 정확도를 나타낸다.

Table 6. Accuracy value.

표 6. 각 데이터에 따른 정확도 측정

Train Data \ Test Data	MNIST	EMNIST (N)	EMNIST (L)	EMNIST (U)	MNIST (P)
MNIST	0.88	0.55	0.09	0.113	0.124
EMNIST (N)	0.569	0.87	0.114	0.086	0.103
EMNIST (L)	0.09	0.076	0.81	0.501	0.083
EMNIST (U)	0.1	0.112	0.326	0.775	0.085
MNIST (P)	0.114	0.096	0.064	0.099	0.885

표 6에서 동일한 Domain을 가지는 데이터를 제외한 다른 데이터는 정확도가 떨어지는 것을 알 수 있다. 그러나 Permuted MNIST의 경우 MNIST와 동일한 Domain을 가지지만 정확도가 낮게 나오는데 이는 상관 계수와 마찬가지로 형태가 유사하지 않기 때문이다. 표 9는 상관 계수와 정확도의 차이를 나타낸다.

표 7에서 형태가 유사하고 Domain이 다른 데이터들은 상관 계수와 정확도의 차이가 상대적으로 크게 나타나는 것을 알 수 있다. 이러한 데이터들은 표 2의 Different Domain 결과처럼 망각현상이 크다는 것을 알 수 있다. Permuted MNIST의 경우 다른 데이터에 대한 상관 계수와 정확도의 차이가 평균 0.529로 비교적 유사한 값을 가진다.

상관 계수와 정확도의 차이가 망각현상의 크기와 비례하는 관계를 바탕으로 망각현상이 심한 데이

Table 7. Difference between Correlation Coefficient and Accuracy.

표 7. 각 데이터에 따른 상관 계수와 정확도 차이

Train Data \ Test Data	MNIST	EMNIST (N)	EMNIST (L)	EMNIST (U)	MNIST (P)
MNIST	0.12	0.378	0.824	0.766	0.594
EMNIST (N)	0.347	0.13	0.745	0.872	0.526
EMNIST (L)	0.78	0.75	0.19	0.432	0.511
EMNIST (U)	0.792	0.843	0.612	0.225	0.549
MNIST (P)	0.399	0.514	0.673	0.463	0.115

터에 대해서 매개변수의 변화량을 억제하면 이전 데이터에 대한 망각현상도 억제시킬 수 있다. 이것을 EWC에 적용하기 위해 비용함수를 살펴봐야 한다. EWC의 비용함수는 식 (1)과 같다.

식 (1)에서 λ 값은 고정된 값으로 이전 Task 데이터의 중요도를 나타낸다. 이 값을 망각현상이 심한 데이터일수록 크게 설정한다면 매개변수의 변화량을 억제할 수 있게 된다. 상관 계수와 정확도의 차이가 망각현상의 크기와 비례하는 관계를 보이므로 λ 값을 이 값과 비례하게 설정한다면 더 망각현상을 더 효율적으로 억제할 수 있을 것이다.

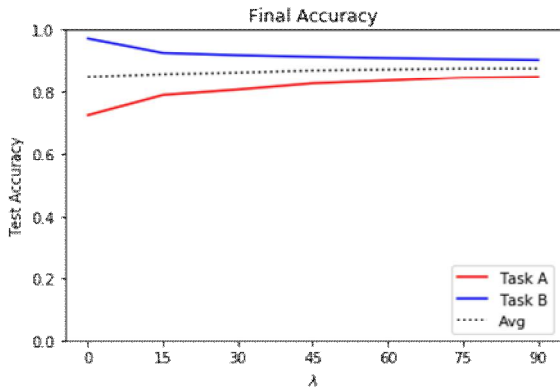
3. 실험 및 고찰

(1) 실험 환경

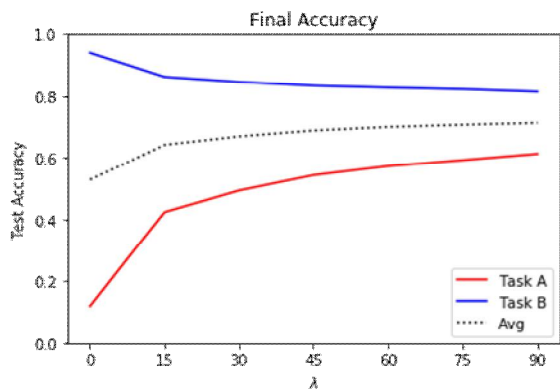
본 논문에서는 MNIST, Permuted MNIST, EMNIST 데이터를 실험에 활용하였다. 표 6과 같은 데이터들의 조합으로 실험을 구성하였고, λ 값은 모든 조합에 대해 0부터 90까지의 값 7가지를 모두 실험하여 비교한다.

첫 번째 Task의 학습 반복 횟수는 800번으로 설정하고 두 번째 Task는 5000번으로 설정하였다. 사용한 모델은 stochastic gradient descent(SGD)를 사용하여 학습하는 일반적인 딥러닝 모델(Vanilla SGD)과 EWC를 사용한 모델을 사용하였다. 신경망 구조는 784, 50, 10개의 노드로 구성된 구조를 사용하였고, Task A의 Learning rate는 0.1로 설정하였고, Task B에서 λ 값이 50을 넘어가는 경우 Loss 함수값이 발산하는 현상이 발생하므로 Task A를 제외한 모든 Learning rate를 0.01로 설정하였다.

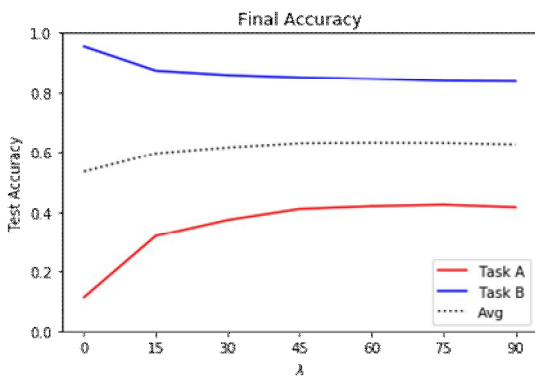
(2) 실험 결과



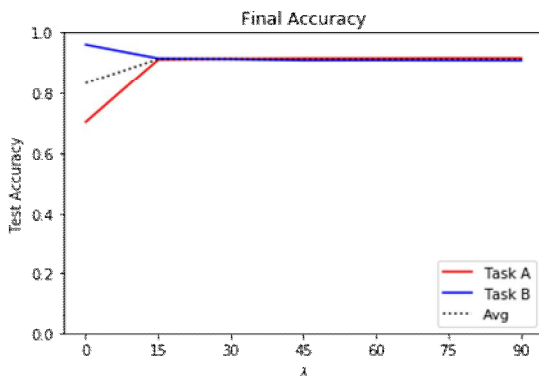
(a) MNIST-EMNIST(Number)



(b) MNIST-EMNIST(Lowercase)



(c) MNIST-EMNIST(Uppercase)



(d) MNIST-Permuted MNIST

Fig. 9. Accuracy Change according to λ .
그림 9. λ 에 따른 최종 정확도 변화량

그림 9는 MNIST 데이터로 학습한 모델을 다양한 데이터로 다시 학습한 후 최종 정확도를 나타낸 그래프이다. 즉, Task A는 MNIST 데이터의 정확도 Task B는 다시 학습한 데이터의 정확도이다. 전체적으로 λ 가 커질수록 Task A의 정확도가 상승하는 모습을 보인다. 하지만 무조건 λ 를 크게 설정한다면 Task B의 정확도가 낮아지게 되고 신경망 학습 시 오차 함수가 발산할 확률이 커지는 문제가 발생할 수 있다. 따라서 적절한 λ 값을 찾기 위해 두 Task의 평균 정확도와 Task A의 정확도 상승 폭이 0.02 이하인 지점의 λ 값을 최적의 값으로 선택하였다. 표 8은 모든 데이터 조합에 대한 최적의 λ 값과 그에 따른 Task A의 정확도 증가량을 나타낸다.

Table 8. Accuracy Increment according to optimal λ .

표 8. 최적 λ 값에 대한 Task A 정확도 증가량

Train Data \ Test Data	Test Data				
	MNIST	EMNIST (N)	EMNIST (L)	EMNIST (U)	MNIST (P)
MNIST	-	0.064	0.454	0.296	0.206
EMNIST (N)	0.051	-	0.241	0.141	0.247
EMNIST (L)	0.283	0.438	-	0.252	0.230
EMNIST (U)	0.296	0.265	0.006	-	0.099
MNIST (P)	0.126	0.123	0.194	0.461	-

상관 계수와 정확도의 차이가 낮은 경우 최적의 λ 값이 잘 변하지 않지만, 차이가 커질수록 급격히 늘어나는 것을 알 수 있다. 따라서 상관 계수와 정확도의 차이를 계산한다면 치명적 망각을 쉽게 일으킬 수 있는 데이터들을 파악하는 기준이 될 수 있다.

III. 결론

CL 환경에서 새로운 학습 데이터가 이전 학습 데이터의 분포와 상반된 분포를 보인다면 성능이 떨어지는 경향을 보였다. 이를 해결하기 위해 이전에 학습된 데이터와 새로 학습할 데이터들의 관계를 파악하는 방법을 제안하였다. 이 방법을 사용하여 이전에 학습한 데이터와 현재 학습할 데이터의

관계를 파악하여 치명적 망각을 일으킬 확률이 높은 데이터를 찾아낼 수 있었고 이 정보를 이용하여 데이터가 학습될 때 미치는 영향을 최소화하기 위해 정규화 강도를 조절하여 학습한 결과 이전 데이터에 대한 정확도의 평균이 약 22.37% 향상되었다. CL 환경에서 인공지능 에이전트를 학습할 때 정제된 데이터들만 입력으로 들어오는 게 아닌 다양한 Domain을 가진 데이터들이 들어올 수 있다. 통일된 Domain이 아닌 다양한 Domain의 데이터들은 학습 모델에 치명적인 망각을 현상을 유발하고 이를 제어하기 위한 본 연구의 방법이 하나의 방법이 될 수 있다. 앞으로는 본 논문의 상관관계를 나타낼 수 있는 효과적인 가변 λ 의 선택으로 능동적인 망각 억제 방법을 제시할 수 있을 것이다.

ACKNOWLEDGEMENTS

This work was supported by Incheon National University (International Cooperative) Research Grant in 2020 (2020-0121)

References

- [1] IJ. Goodfellow, et al., “An empirical investigation of catastrophic forgetting in gradient-based neural networks,” *arXiv preprint arXiv:1312.6211*, 2013.
- [2] RM. French, “Catastrophic forgetting in connectionist networks,” *Trends in cognitive sciences*, Vol.3, No.4, pp.128-135, 1999.
DOI: 10.1016/S1364-6613(99)01294-2
- [3] J. Kirkpatrick, et al., “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the national academy of sciences*, Vol.114, No.13, pp.3521-3526, 2017. DOI: 10.1073/pnas.1611835114
- [4] YC. Hsu, et al., “Re-evaluating continual learning scenarios: A categorization and case for strong baselines,” *arXiv preprint arXiv:1810.12488*, 2018.
- [5] V. Lomonaco, and D. Maltoni, “Core50: a new dataset and benchmark for continuous object recognition,” *arXiv preprint arXiv:1705.03550*, 2017.
- [6] D. Maltoni, and V. Lomonaco, “Continuous

learning in single-incremental-task scenarios,” *Neural Networks*, Vol.116, pp.56-73, 2019.

DOI: 10.1016/j.neunet.2019.03.010

[7] Y. LeCun, et al., “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, Vol.86, No.11, pp.2278-2324, 1998.

DOI: 10.1109/5.726791

[8] G. Cohen, et al. “EMNIST: Extending MNIST to handwritten letters,” In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp.2921-2926, 2017.

DOI: 10.1109/IJCNN.2017.7966217

[9] GL. Parisi, et al., “Continual lifelong learning with neural networks: A review,” *Neural Networks*, 2019. DOI: 10.1016/j.neunet.2019.01.012.

[10] R Kemker, et al, “Measuring catastrophic forgetting in neural networks,” In: *Thirty-second AAAI conference on artificial intelligence*. 2018.

BIOGRAPHY

Seong-Hyeon Park (Member)



2020~ : MS degree in Embedded Systems Engineering, Incheon National University.
A.I. Deep Learning, Embedded Systems Engineering

Seok-Hoon Kang (Member)



1989 : BS degree in Electronic Communications Engineering, Hanyang University.
1995 : PhD degree in Electronic Communications Engineering, Hanyang University.
2004~ : Professor, Embedded Systems Engineering, Incheon National University
A.I., Deep Learning, Mobile/Embedded System, Wearable System, Natural Language Processing