

지속적 학습 환경에서 효율적 경로 선택

Efficient Path Selection in Continuous Learning Environment

박 성 현*, 강 석 훈*

Seong-Hyeon Park*, Seok-Hoon Kang*

Abstract

In this paper, we propose a performance improvement of the LwF method using efficient path selection in Continuous Learning Environment. We compare performance and structure with conventional LwF. For comparison, we experiment with performance using MNIST, EMNIST, Fashion MNIST, and CIFAR10 data with different complexity configurations. Experiments show up to 20% improvement in accuracy for each task, which mitigating the Catastrophic Forgetting phenomenon in Continuous Learning environments.

요 약

본 논문에서는, 지속적 학습 환경에서 효율적 경로 선택에 의한 LwF방법의 성능향상을 제안한다. 이를 위해 컨볼루션 레이어를 분리하는 방법을 사용하여 기존의 LwF와 성능 및 구조를 비교한다. 비교를 위해 복잡도가 다른 구성을 가진 MNIST, EMNIST, Fashion MNIST, CIFAR10 데이터를 사용하여 성능을 실험하였다. 실험결과, 각 태스크 별 정확도가 최대 20% 향상되었으며, LwF 기반의 지속적 학습 환경에서 치명적 망각 현상이 개선되었다.

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

1. 서론

LwF(Learning without Forgetting)는 지속적 학습(Continual Learning) 환경에서 발생하는 과거 데이터 지식의 망각(치명적 망각, Catastrophic Forgetting) 현상을 완화시키기 위한 방법들 중 하나이다[1, 2, 3, 4]. 이 방법은 마지막 출력 레이어를 제외한 모든 레이어를 공유하는 구조를 갖고 있고, 학습된 결과를 배포하기 용이한 특징을 가지고 있다. LwF는 과거에 학습한 데이터와 새롭게 학습할 데이터의 도메인이 동일하거나 유사한 경우 좋은

성능을 보이지만, 도메인이 다르거나 유사도가 낮은 경우, 그렇지 않은 경우에 비해 성능이 떨어지는 모습을 보인다[1]. 이는 다른 정규화 방법들에서도 동일하게 나타나는 현상이고[5], 해결해야 할 문제이다.

앞으로 학습할 도메인에 대해 높은 수준의 내부 구조(high degree of internal structure)를 갖는 도메인으로 사전 학습을 진행한 경우 망각 현상은 잘 발생하지 않는다[6]. 이와 반대로 도메인이 유사하지 않아 이전 태스크(Task)로 학습된 가중치들로 구성된 내부 구조가 새로운 태스크를 설명하기에

* Dept. of Embedded Systems Engineering, Incheon National University

★ Corresponding author

E-mail : hana@inu.ac.kr, Tel : +82-32-835-8760

※ Acknowledgment

This work was supported by Incheon National University (International Cooperative) Research Grant in 2021 (2021-0089) Manuscript received Jul. 29, 2021; revised Sep. 14, 2021; accepted Sep. 14, 2021.

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.

부족한 정보를 가지고 있는 경우 학습 과정에서 새로운 태스크를 잘 설명할 수 있는 방향으로 구조가 변경된다. 이때 공유 가중치에 급격한 변화가 발생하게 되고 이 변화로 인해 망각 현상이 더 크게 발생한다[1].

공유 가중치의 급격한 변화를 막기 위해 공유하는 영역을 분리하여 각 태스크가 공유 가중치에 미치는 영향을 줄인다면 망각 현상이 완화되는 효과를 볼 수 있을 것이다. 따라서 유사도가 낮은 데이터가 들어온 경우 모든 레이어의 가중치를 공유하지 않고 일부를 분리하는 방법을 생각할 수 있다.

EWC의 실험 결과에서는, 데이터의 유사도가 낮아질수록 출력단과 가까운 하위 레이어의 공유 비율은 유지되는 반면 입력단과 가까운 상위 레이어의 공유 비율은 낮아지게 된다[7]. 이 실험 결과는 하위 레이어의 가중치들은 각 태스크들을 설명하기에 충분한 정보를 가지고 있지만 상위 레이어의 가중치들은 부족한 정보를 가지고 있다고 설명할 수 있다. 이러한 관점으로 보았을 때 하위 레이어는 충분한 정보를 가지고 있으므로 학습 과정에서 구조의 변동이 적어 망각 현상이 상대적으로 적게 발생할 것이고, 상위 레이어는 정보의 부족으로 인해 구조의 변동이 심해 망각 현상이 상대적으로 많이 발생할 것이다.

본 논문에서는 하위 레이어인 FC 레이어를 분리하는 방법보다, 상위 레이어인 컨볼루션 레이어를 분리하고, 이를 효율적으로 선택하는 방법을 사용하여 치명적 망각 현상을 완화시키고, 정확도를 높이는 방법을 제안한다. 본 논문의 실험과 방법은 싱글 태스크가 연속적으로 추가되는 지속적 학습 환경을 대상으로 한다.

II. 본론

1. 관련 연구

가. Learning without Forgetting (LwF)

Learning without Forgetting (LwF)는[1], 지식 증류(knowledge distillation)를 사용하여[8], 이전 태스크로 학습된 모델의 출력 분포를 유지하여 치명적 망각 현상을 완화하는 방법이다. LwF의 손실 함수는 (1)과 같이 나타낼 수 있다.

$$L = L_n(Y_n, \hat{Y}_n) + \lambda L_o(Y_o, \hat{Y}_o) + L_2 \quad (1)$$

$L_o(Y_o, \hat{Y}_o)$ 는 모델의 출력 분포를 유지하기 위한 손실함수 값이다. Y_o 는 이전 태스크로 학습된 모델에 새로운 태스크를 입력하여 만들어낸 출력 분포이고 이 값을 모델의 출력 \hat{Y}_o 의 목표로 설정하여 과거 출력 분포를 유지하게 한다. $L_n(Y_n, \hat{Y}_n)$ 는 현재 태스크를 위한 크로스 엔트로피 손실 함수이다. L_2 은 L2 정규화 값, λ 는 이전 태스크에 대한 정규화 강도이다.

나. Elastic Weight Consolidation (EWC)

Elastic Weight Consolidation (EWC)는[7], 과거 태스크와 깊게 관련되어 있는 가중치들의 변화량을 손실함수에 추가하여 관련된 가중치를 잘 변화하지 않게 하는 방법을 사용해 치명적 망각 현상을 완화한다. EWC의 손실함수는 (2)와 같이 나타낼 수 있다.

$$L = L_n + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta_{o,i}^*)^2 \quad (2)$$

L_n 는 현재 태스크를 위한 크로스 엔트로피값, F_i 는 과거 태스크와 가중치가 얼마나 연관되어 있는지 나타내는 중요도, λ 는 이전 태스크에 대한 정규화 강도이다.

다. Synaptic Intelligence (SI)

Synaptic Intelligence (SI)는[9], EWC와 유사한 방법이다. 이전 태스크와 관련되어 있는 가중치들을 선택할 때 학습 과정에서 손실함수의 변화량을 기준으로 큰 영향을 미친 가중치들을 선택한다. SI의 손실함수는 (3)과 같이 나타낼 수 있다.

$$L = L_n + \lambda \sum_i \Omega_i (\theta_i - \theta_{o,i}^*)^2 \quad (3)$$

Ω_i 는 손실함수의 변화량을 기준으로 가중치들이 변화량에 얼마나 영향을 끼쳤는지 나타내는 중요도, λ 는 이전 태스크에 대한 정규화 강도이다.

2. 본 논문의 레이어 분리 및 효율적 선택 방법

본 논문에서는, LwF에서, 태스크 마다 공유 비율이 높은 FC 레이어를 분리하는 방법 대신, 공유 비율이 낮은 상위 레이어인 컨볼루션 레이어를 분리시키고, 태스크마다 레이어를 효율적으로 선택하는

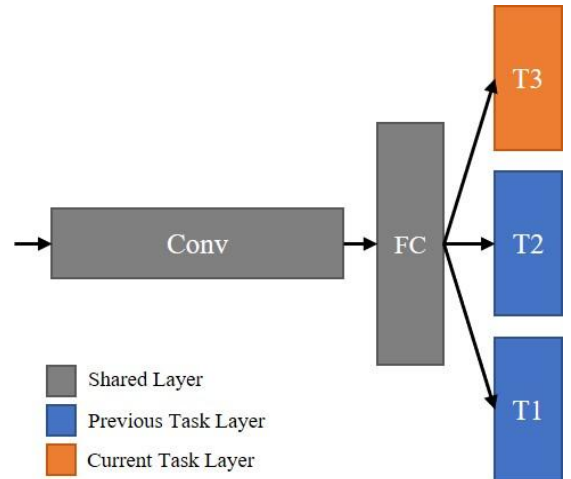
방법을 제안한다. 모델의 가중치 공유 비율이 낮은 부분을 분리하여, 모델의 표현능력과 효율성을 높이고, 이를 통해 이전 학습내용을 좀 더 잘 유지하게 한다. 레이어를 분리할 때 모든 컨볼루션 레이어를 분리하게 하면, 새로운 태스크를 위한 파라미터 공간이 넓어지므로 좋은 방법이 될 수도 있다. 하지만 기존 컨볼루션 레이어에 저장된 이전 태스크의 지식을 사용하지 않는 방법이므로 효율적인 방법이라고 보기 어렵다. 본 논문에서는 신경망의 활성화 출력값을 이용하고, 두 태스크 사이의 활성화 출력값을 각 컨볼루션 레이어 별로 측정하고, 유사도가 높은 경우 레이어를 공유하고 유사도가 낮은 경우 레이어를 분리하는 방법을 사용하는 방법을 제안한다. 이렇게 하면, 각 레이어에 존재하는 기존 태스크의 지식을 적절하게 활용할 수 있어서[10], 레이어를 효율적으로 사용하고, 학습결과도 공유할 수 있게 된다.

또한, 본 논문에서는, 제안하는 방법을, 기존의 LwF와 비교하고, LwF의 컨볼루션 레이어를 단순히 분리한 경우와도 비교 실험하여, 그 성능을 비교한다. 레이어를 분리하는 것만으로 발생하는 성능의 차이를 비교할 수 있다.

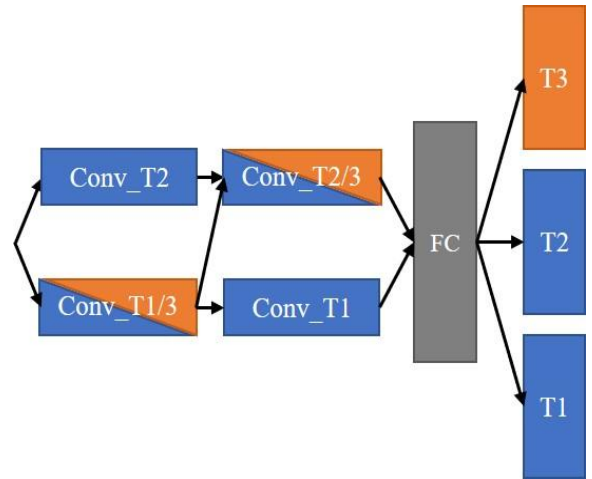
기존 LwF와 본 논문에서 제안하는 LwF의 컨볼루션 레이어 분리 및 선택 방법을 그림 1.에 나타내었다. 그림 1.의 (a)는 LwF의 구조를 나타낸다. 기본적인 신경망 구조를 기반으로 태스크마다 독립적인 출력 레이어가 존재하는 구조이다. (b)는 본 논문에서 제안하는 방법으로, 레이어 유사도에 따라 컨볼루션 레이어를 부분적으로 공유하는 방법이다. 각 태스크가 들어올 때마다 유사도가 높은 레이어를 선택해 학습하고 그 외의 레이어들은 학습이 진행되지 않도록 한다.

본 논문에서는, 각 태스크가 들어올 때 마다, 컨볼루션 레이어를 단순히 분리하며, FC 레이어는 공유하고, 태스크마다 독립적인 컨볼루션 레이어를 갖게 하여, 태스크가 추가될 때마다, 새롭게 추가된 컨볼루션 레이어로 학습을 진행하여, 각 태스크는 다른 컨볼루션 레이어와 학습이 진행되지 않도록 하는 방법도 진행하고, 이 결과를 본 논문의 결과와 비교 실험한다.

표 1.에 실험에 사용할 모델의 구성과 각 레이어마다, 본 논문의 실험에 필요한 가중치 수를 나타내었다.



(a) LwF.



(b) Convolution layer select.

Fig. 1. Model Structure Comparison.

그림 1. 모델 구조의 비교

Table 1. Model Description in this paper.

표 1. 실험에 사용할 모델의 구성

Name	Output Dim	Filter Size	Weight #
Conv1	30×30×32	3×3	896
Conv2	28×28×64	3×3	18496
Pool1	14×14×64	2×2	-
FC1	128	-	165760
FC2	10	-	1290

표 1.의 모델을 사용하는 환경에서 하위 레이어인 FC 레이어를 분리하여 실험할 경우, 각 태스크마다, 추가적인 1,605,760개의 가중치가 생성되게 된다. 이는 전체 모델의 약 98%를 차지하는 크기이다. 이 방법으로 진행하게 된다면 태스크가 진행될수록 가중치가 차지하는 메모리가 크게 증가하게 된다. 또한 기존 LwF에서 FC 레이어를 분리하

는 실험을 수행하였으나, 늘어나는 가중치 수에 비해 성능 향상은 많지 않았다는 것을 이미 밝히고

Table 2. Algorithm of this paper.

표 2. 본 논문의 알고리즘

Algorithm 1. Basic LwF
Input : Dataset $D=(D_1, \dots, D_T)$, λ Train for $t=1, \dots, T$ do if $t=1$ then Train the network using normal cross entropy loss else Make Y_o for old Tasks using the network and D_t Make output layer for Task t Train the network using Eq. 1
Algorithm 2. Convolution layer split
Input : Dataset $D=(D_1, \dots, D_T)$, λ Train for $t=1, \dots, T$ do if $t=1$ then Train the network using normal cross entropy loss else Make Y_o for old Tasks using the network and D_t Split the Convolution layer and assign new Task Freeze other Convolution layers Make output layer for Task t Train the network using Eq. 1
Algorithm 3. Convolution layer select
Input : Dataset $D=(D_1, \dots, D_T)$, λ , Threshold τ Train for $t=1, \dots, T$ do if $t=1$ then Train the network using normal cross entropy loss else Make Y_o for old Tasks using the network and D_t for $l=1, \dots, L$ then Calculate layer similarity S by using [10] if $S > \tau$ then Select the layer l for Task t else Split the Convolution layer l for Task t Freeze all Convolution layers that are not associated with Task t Make output layer for Task t Train the network using Eq. 1

있다[1].

콘볼루션 레이어는, 학습할 때, 많은 연산량이 요구된다. 따라서 학습 시간과 학습에 사용되는 메모리가 늘어날 수 있다. 그러나 896+18496개의 가중치만 추가로 생성된다. 이는 전체 모델의 약 1%를 차지하는 크기이다. 따라서 LwF의 특징인 학습된 모델을 배포할 때, 본 논문의 방식인, 콘볼루션 레이어 분리 및 선택 방법이 효율적인 배포 방법이 될 수 있다.

표 2.에 본 논문에서 사용하는 레이어 분리와 선택 알고리즘(알고리즘 3, 이하 “Convolution Layer Select” 방법)을 나타내었다. 아울러, 비교실험을 위해, LwF의 기본 방법(알고리즘 1, 이하 “Basic LwF” 방법)과 단순 콘볼루션 레이어 분리에 의한 비교 실험 방법(알고리즘 2, 이하 “Convolution Layer Split” 방법)도 같이 나타내었다.

첫 번째 알고리즘은 기본 LwF 방법으로 첫 번째 태스크는 일반적인 교차 엔트로피 Loss를 사용하여 학습을 진행하고 이후 태스크는 새로운 데이터를 사용하여 출력 분포 Y_o 를 만들어 과거 태스크 출력 레이어의 목표 값으로 설정하여 수식 (1)을 사용하여 학습을 진행한다. 두 번째 알고리즘은 새로운 태스크가 들어올 때마다 콘볼루션 레이어를 분리하는 방법을 나타낸다. 기본 LwF 방법과의 차이점은 모든 태스크가 독립된 콘볼루션 레이어를 가지고 있고 현재 태스크를 학습하는 중에는 다른 태스크들의 콘볼루션 레이어의 가중치들이 고정되어 학습되지 않는 점이 있다. 세 번째 알고리즘은, 본 논문에서 제안하는 방법으로, 새로운 태스크가 들어올 때마다 무조건 새 콘볼루션 레이어를 할당하는 것이 아닌 레이어 별 유사도를 측정하여 유사도가 높은 레이어는 공유하도록 하는 방법이다. 두 번째 방법과 마찬가지로 현재 태스크와 연관되지 않는 콘볼루션 레이어들의 가중치는 고정되어 학습되지 않는다.

3. 실험 환경

지속적 학습 환경에서 발생하는 치명적 망각 현상은 태스크 데이터 사이의 유사도가 낮아질수록 강도가 심해지는 경향을 보인다[1, 9]. 따라서, 이를 고려하여 망각현상이 심하지 않은 경우와 심한 경우로 구분 지어 실험 시나리오를 구성하여야 한다. 본 논문에서 사용할 데이터는 Mnist, Emnist, Fashion

mnist, Cifar10 데이터이고, 학습할 데이터의 복잡도에 따라, 6가지 시나리오로 학습을 구성하였다. 이로 인해 본 논문의 방법이 각 데이터의 특성에 따라 어떻게 개선되는지를 알 수 있게 된다.

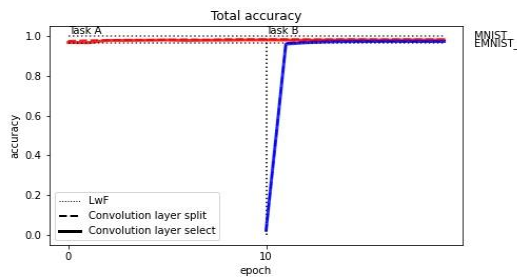
표 3.의 실험 시나리오는, 태스크끼리의 복잡도를 기반으로 나눈 것이다. 시나리오 1은 복잡도가 낮고 유사한 데이터를 사용하여 태스크를 구성하였고, 시나리오 2는 복잡도가 높고 유사하지 않은 데이터를 사용한 경우이다. 시나리오 3, 4는 복잡도, 유사도의 차이가 큰 경우이고, 시나리오 5, 6은 서로 다른 복잡도와 유사도를 가진 데이터가 연속적으로 들어오는 경우이다. 단순한(학습하기 쉬운) 데이터로는 Mnist, Emnist 데이터를 할당하였고, 복잡한 데이터(상대적으로 학습하기 어려운)로는 Fashion mnist, Cifar10 데이터를 할당하였다. 학습 batch 크기는 100개, epoch는 각 태스크 별로 10회로 설정하였고, 모델은 기본 LwF 방법을 사용하는 모델과 컨볼루션 레이어 분리 방법을 적용한 모델들을 사용하여 비교하였다. LwF의 λ 값은 15.0으로 설정하였다.

Table 3. Experiment Scenario.

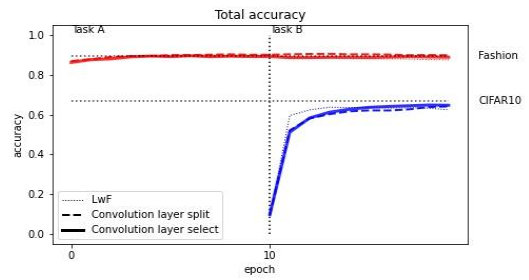
표 3. 시나리오 구성

Senario	Task A		Task B	
1	Mnist		Emnist(eng)	
2	Fashion		Cifar10	
3	Mnist		Cifar10	
4	Cifar10		Mnist	
	Task A	Task B	Task C	Task D
5	Mnist	Emnist(up)	Fashion	Cifar10
6	Cifar10	Fashion	Emnist(up)	Mnist

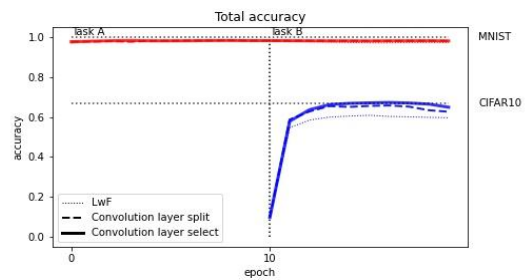
4. 실험 결과



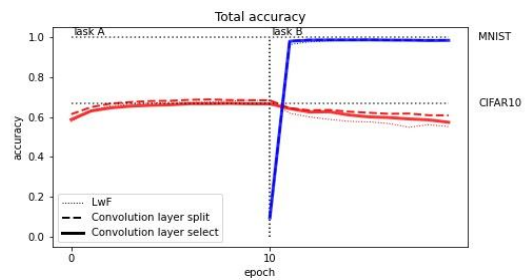
(a) Scenario 1.



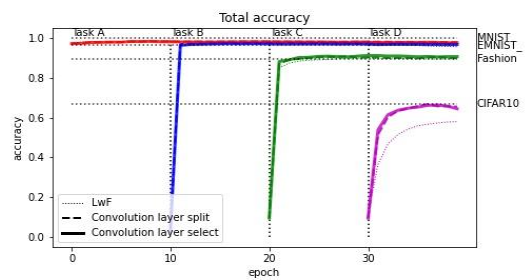
(b) Scenario 2.



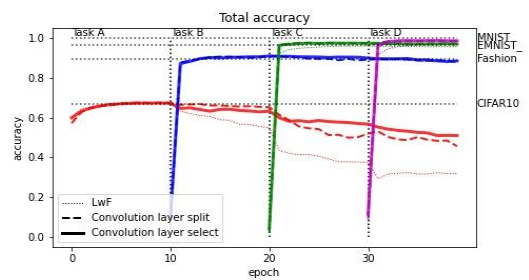
(c) Scenario 3.



(d) Scenario 4.



(e) Scenario 5.



(f) Scenario 6.

Fig. 2. Experiment Result.

그림 2. 실험 결과

Table 4. Final accuracy for each Task (%).

표 4. 태스크 별 최종 정확도 (%)

Senario		Task A		Task B	
1	LwF	97.9		97.2	
	+Split	98.1		97.3	
	+Select	97.9		97.1	
2	LwF	87.4		62.3	
	+Split	89.7		64.0	
	+Select	88.7		64.6	
3	LwF	97.1		59.6	
	+Split	97.9		62.5	
	+Select	97.9		64.8	
4	LwF	55.2		98.5	
	+Split	60.7		98.3	
	+Select	57.3		98.2	
		Task A	Task B	Task C	Task D
5	LwF	97.7	95.5	89.3	57.9
	+Split	97.7	97.1	89.6	65.4
	+Select	97.5	96.5	90.5	64.4
6	LwF	31.8	88.5	95.6	97.1
	+Split	45.6	87.7	97.0	98.3
	+Select	50.9	88.3	96.8	98.4

그림 2와 표 4.에, 표 3.의 시나리오에 따른 실험 결과를 보이고 있다. 그림 2.는 학습이 진행될수록 변화하는 태스크 정확도를 나타낸다. 얇은 점선은 기본 LwF 방법(LwF)을, 굵은 점선은 단순 콘볼루션 레이어 분리 방법(+Split)을, 굵은 실선은 본 논문의 방법인, 콘볼루션 레이어 부분 분리 및 선택 방법(+Select)을 나타낸다. 단순 콘볼루션 레이어 분리 방법은, 본 논문의 방법에 대한 비교 실험이다.

표 4.의 시나리오 별 최종 정확도를 분석해 보면, 시나리오 1에서 복잡도가 낮고 유사한 데이터들은 모든 방법에서 눈에 띄는 정확도 하락을 보이지 않는다는 것을 알 수 있다. 시나리오 2에서 본 논문의 방법은, 기본 LwF와 비교해 모든 태스크에서 정확도가 약 1~2% 향상된 결과를 보인다. 시나리오 3에서, 본 논문의 방법이 기본 LwF 방법보다 약 5% 향상된 결과를 보이고 있다. 이것은 단순 콘볼루션 레이어 분리 방법이 약 3% 향상된 결과를 보인 것 보다 더 나은 결과를 나타낸 것이다. 시나리오 4에서 Task A 정확도를 확인하면 약 2%~5% 향상된 결과를 보인다. 시나리오 5, 6 모두 복잡한

데이터인 Cifar10의 정확도가 약 7~20% 성능 향상을 보인다. 특히 시나리오 6의 Cifar10(Task A) 정확도의 경우, 본 논문의 방법에서 약 20% 향상되어, 망각 현상이 현저하게 개선된 것을 확인할 수 있다.

표 4.의 Scenario 1~4의 결과를 보면, mnist, emnist와 같은 단순한 데이터들과 달리, Cifar10과 같은 복잡도가 높은 데이터의 성능이 약 5% 향상되어(시나리오 3, Task B), 다른 데이터들 보다 더 높은 정확도 상승치를 보인다. 이는 콘볼루션 레이어를 분리하는 방법이 복잡도가 높은 데이터들의 지식을 유지하는데 도움을 줄 수 있음을 나타낸다. 표 4.의 시나리오 5, 6에서, 후반 태스크가 진행될수록 본 논문의 방식은 기존 LwF 방법에 비해 정확도가 높게 측정된다. 시나리오 5에서는 모든 태스크의 정확도가 상승하였고, 특히 Task D의 정확도가 약 7% 향상되어 좋은 성능을 보인다. 시나리오 6에서는 Task A의 정확도 하락 폭이 약 7% 줄어들어 기존 태스크의 지식을 보존하는 모습을 보인다. 이는 본 논문의 콘볼루션 레이어 분리 및 선택 방법이 많은 태스크가 진행되는 환경에서도 효과가 있음을 나타내는 것이다. 학습이 어려워지는 현상을 완화하고 오래된 태스크의 정확도 유지에 도움을 줄 수 있는 것이다.

Table 5. Added Weight number.

표 5. 추가된 weight 개수

Senario	+Split	+Select
1		0
2	19K	18K
3		19K
4		19K
5	57K	36K
6		56K

표 5.는 레이어 분리 및 선택을 위해, 본 논문의 실험에서 추가한 가중치 개수를 보이고 있다. 단순 레이어 분리 방법(+Split)과, 본 논문의 방법(+Select)의 시나리오별 개수 차이를 확인할 수 있다. 콘볼루션 레이어 단순 분리 방법은, 태스크가 들어올 때마다 모든 콘볼루션 레이어를 추가하게 된다. 이 방법을 사용하면 시나리오 1~4은 1개의 태스크가 추가로 들어오는 환경이므로, 모든 콘볼루션 레이어

어의 가중치 개수인 약 19K개의 가중치가 추가되고, 시나리오 5, 6은 3개의 태스크가 추가되므로, 약 57K개의 가중치가 추가되게 된다.

본 논문의 방법은, 태스크 사이에 유사도가 높은 레이어를 선택하여 공유하는 방법을 사용하므로, 추가되는 가중치를 줄일 수 있다. 시나리오 3, 4를 제외한 모든 시나리오에서 추가되는 가중치가 줄어든 것을 확인할 수 있다.

표 4의 시나리오 1의 결과를 보면, 실험 태스크 데이터들의 복잡도가 낮고 유사하므로 추가적인 콘볼루션 레이어의 분리 없이도 좋은 성능을 보이는 것을 확인할 수 있다. 이러한 환경에서 모든 콘볼루션 레이어를 분리하는 방법은 불필요한 오버헤드를 일으키므로, 레이어의 유사도를 파악해 기존에 사용한 레이어를 재사용하는 본 논문의 방법이 더 효율적이다. 마찬가지로 시나리오 5의 결과를 보면 콘볼루션 레이어 부분 분리 방법의 성능은 모든 레이어 분리 방법과 비슷하지만 추가되는 가중치는 약 40% 감소하므로 더 효율적이다. 또한 시나리오 6의 Task A의 경우 추가되는 가중치는 더 적지만 오히려 정확도가 상승한 것을 확인할 수 있다.

III. 결론

본 논문에서는, LwF에서 공유 비율이 높은 FC 레이어를 분리하는 방법 대신, 공유 비율이 낮은 상위 레이어인 콘볼루션 레이어를 분리시키는 방법을 사용해 모델의 가중치 공유 효율성을 높여 이전 태스크의 학습 내용을 보존하고자 하였다. 이 방법을 사용하여 LWF 방법보다 각 태스크의 정확도가 최대 약 20% 상승한 것을 확인할 수 있었다. 이는 지속적 학습 환경에서 치명적 망각 현상이 그만큼 줄어드는 것을 의미하는 것이다. 콘볼루션 레이어를 분리시키는 방법은 연산량을 증가시키므로 메모리 사용량과 학습시간이 늘어날 수 있다. 그러나 FC 레이어 분리 방법보다 가중치 개수가 약 98%(표 1. 참고) 감소하므로, 모델의 배포가 용이해지는 장점이 있다. 추가적으로 망각 현상을 심하게 일으키지 않는 태스크가 들어올 때에도 불필요하게 모든 콘볼루션 레이어를 분리하는 비효율성을 해결하기 위해 유사도에 따라 분리할 레이어를 선택하는 알고리즘을 제안하였고, 그 결과 정확도

를 유지하며 추가되는 가중치를 최대 40% 감소시킬 수 있었으며, 모든 콘볼루션 레이어를 분리하는 방법보다 정확도가 최대 5% 상승한 결과도 확인할 수 있었다.

ACKNOWLEDGEMENTS

This work was supported by Incheon National University (International Cooperative) Research Grant in 2021 (2021-0089)

References

- [1] Z. Li, and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, Vol.40, No.12, pp.2935-2947, 2017. DOI: 10.1109/TPAMI.2017.2773081
- [2] IJ. Goodfellow, et al., "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211*, 2013.
- [3] GI. Parisi, et al., "Continual lifelong learning with neural networks: A review," *Neural Networks*, Vol.113, pp.54-71, 2019. DOI: 10.1016/j.neunet.2019.01.012
- [4] 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
- [5] YC. Hsu, et al., "Re-evaluating continual learning scenarios: A categorization and case for strong baselines," *arXiv preprint arXiv:1810.12488*, 2018.
- [6] K. McRae, and PA. Hetherington, "Catastrophic interference is eliminated in pretrained networks," In: *Proceedings of the 15h Annual Conference of the Cognitive Science Society*, pp.723-728, 1993.
- [7] 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.
- [8] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS*

Workshop, 2014.

[9] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp.3987-3995, 2017.

[10] SH. park, SH. Kang, "Continual Learning using Data Similarity," *Journal of IKEEE*, Vol.24, No.2, pp.514-522, 2020.

DOI: 10.7471/ikeee.2020.24.2.514

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.