

LwF에서 망각현상 개선을 위한 적응적 가중치 제어 방법

박성현¹ · 강석훈^{2*}

Adaptive Weight Control for Improvement of Catastrophic Forgetting in LwF

Seong-Hyeon Park¹ · Seok-Hoon Kang^{2*}

¹Graduate Student, Department of Embedded Systems Engineering, Incheon National University, Incheon, 22012 Korea

^{2*}Professor, Department of Embedded Systems Engineering, Incheon National University, Incheon, 22012 Korea

요 약

지속적 학습 환경을 위한 학습 방법 중 LwF(Learning without Forgetting)는 정규화 강도가 고정되어 있어 다양한 데이터가 들어오는 환경에서 성능이 하락할 수 있다. 본 논문에서는 학습하려는 데이터의 특징을 파악하여 가중치를 가변적으로 설정할 수 있는 방법을 제안하고, 실험으로 성능을 검증한다. 상관 관계와 복잡도를 이용하여 적응적으로 가중치를 적용하도록 하였다. 평가를 위해 다양한 데이터를 가진 태스크가 들어오는 시나리오를 구성하여 실험을 진행하였고, 실험 결과 새로운 태스크의 정확도가 최대 5%, 이전 태스크의 정확도가 최대 11% 상승하였다. 또한, 본 논문에서 제안한 알고리즘으로 구한 적응적 가중치 값은, 각 실험 시나리오마다 반복적 실험에 의해, 수동으로 계산한 최적 가중치 값에 접근한 것을 알 수 있었다. 상관 계수 값은 0.739 이었고, 전체적으로 평균 태스크 정확도가 상승하였다. 본 논문의 방법은, 새로운 태스크를 학습할 때마다 적절한 람다 값을 적응적으로 설정하였으며, 본 논문에서 제시한 여러 가지 시나리오에서 최적의 결과값을 도출하고 있다는 것을 알 수 있다.

ABSTRACT

Among the learning methods for Continuous Learning environments, "Learning without Forgetting" has fixed regularization strengths, which can lead to poor performance in environments where various data are received. We suggest a way to set weights variable by identifying the features of the data we want to learn. We applied weights adaptively using correlation and complexity. Scenarios with various data are used for evaluation and experiments showed accuracy increases by up to 5% in the new task and up to 11% in the previous task. In addition, it was found that the adaptive weight value obtained by the algorithm proposed in this paper, approached the optimal weight value calculated manually by repeated experiments for each experimental scenario. The correlation coefficient value is 0.739, and overall average task accuracy increased. It can be seen that the method of this paper sets an appropriate lambda value every time a new task is learned, and derives the optimal result value in various scenarios.

키워드: 지속적 학습, 치명적 망각, 신경망, 딥러닝, 정규화

Keywords: Continuous learning, Catastrophic forgetting, Neural network, Deep learning, Regularization

Received 6 September 2021, Revised 9 September 2021, Accepted 30 September 2021

* Corresponding Author Seok-Hoon Kang(E-mail:hana@inu.ac.kr, Tel:+82-32-835-8422)

Professor, Department of Embedded Systems Engineering, Incheon National University, Incheon, 22012 Korea

Open Access <http://doi.org/10.6109/jkiice.2022.26.1.15>

print ISSN: 2234-4772 online ISSN: 2288-4165

© 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.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

다양한 특징을 가진 데이터들을 순차적으로 입력받아 학습하는 지속적 학습(Continuous Learning, CL) 환경에서는[1], 치명적인 망각(Catastrophic Forgetting) 현상이 발생할 수 있다[2, 3]. 이 현상은 학습 태스크(Task)들이 순차적으로 들어오고 학습이 진행될수록 과거에 학습한 태스크들의 정보가 점차 사라지게 되는 것이다. 이러한 현상을 해결하기 위해서 사용하는 방법들 중 Learning without Forgetting(LwF)[3, 4]는, 과거에 학습한 태스크들의 출력값을 유지하여 망각현상을 최소화하는 방법을 사용하고 있다.

LwF에는 과거의 태스크 출력값을 얼마나 강하게 유지할지 결정하는 가중치(λ)가 존재한다. 이는 학습과정을 수행하기 전에 사용자가 임의의 상수로 결정하는 값이다. 이 가중치 값을 크게 설정할수록 과거 태스크 출력값을 유지하려는 제약이 커지게 되고, 학습 과정에서 이 출력값과 연관된 가중치들의 변화가 최소화되어 과거 태스크들의 지식이 보존된다. 과거 태스크의 성능 하락을 최소화하기 위해 단순히 과거 학습 태스크에 가중치를 높게 설정하여 학습하는 방법을 사용할 수 있지만, 반대로 이는 현재 학습 태스크의 성능 저하로 이어질 수 있는 단점이 된다. 만약 학습하려는 데이터의 특징을 파악하여 과거 지식의 망각 위험이 큰 데이터인 경우 가중치를 높게 설정해 과거 태스크들의 지식을 보존하고 망각 위험이 적은 데이터인 경우 가중치를 낮게 설정해 현재 태스크들의 지식 습득에 가중을 두는 방법을 사용하면 가중치가 고정되어 있는 기존 방법의 장단점을 보완할 수 있을 것이다. 따라서 본 논문에서는, LwF 방법에서, 망각 가능성을 파악하고, 이것에 따라 가중치를 태스크에 대해 적응적으로 설정할 수 있게 하여, LwF가 가지는 망각현상을 줄일 수 있는 방법을 제안한다.

II. 본론

2.1. 지속적 학습

지속적 학습 환경은 다양한 환경들이 존재하며, 대체적으로 다음과 같이 분류가 가능하다[5].

- Incremental Task Learning

- Incremental Domain Learning
- Incremental Class Learning

Incremental Task Learning 환경은 각 태스크마다 새로운 출력 레이어가 할당되는 환경으로 각 태스크들은 서로 독립되어 있는 형태이다. 일반적으로 이 방법은 여러 데이터 세트(Mnist, Cifar10 등)를 사용하여 수행된다. LwF와 같은 방법들은 이 환경에서 실험을 수행한다. Incremental Domain Learning 환경은 클래스가 고정된 상태에서 클래스들의 데이터가 순차적으로 들어오는 환경이다. 이 때 각 태스크들은 동일한 출력 레이어를 공유하며 학습이 진행될수록 클래스들의 도메인(Domain)이 확장되는 특징이 있다. 일반적으로 각 태스크 데이터들은 하나의 데이터 세트나 서로 유사도가 높은 데이터를 사용한다. Elastic Weight Consolidation(EWC)[3, 6]와 같은 방법들은 이 환경에서 실험을 수행한다. Incremental Class Learning 환경은 각 태스크마다 새로운 클래스가 추가되는 환경으로 각 태스크들의 출력 레이어는 공유된다. 이 환경은 태스크들의 구분이 필요 없으며 다른 방법에 비해 난이도가 어려운 환경이다.

본 논문은 LwF를 기반으로 하고 있으므로, Incremental Task Learning 환경에서 실험하게 되며, 표 1과 같은 데이터 셋들을 태스크로 설정한다.

2.2. 관련 연구

치명적인 망각을 줄이기 위한 방법들은 다양한 방법들이 여러 가지 발표되고 있으며, 현재까지 발표된 방법은 크게 3가지 갈래로 분류가 가능하다.

2.2.1. 정규화 방법

정규화 방법은 이전에 학습된 데이터와 연관된 가중치들의 변화를 억제하는 방법을 사용한다. LwF 방법은 과거 학습 데이터로 학습했던 모델의 출력값을 유지하여 과거 학습 데이터의 지식을 우회적으로 보존하는 방법을 사용한다. EWC 방법은 과거 학습 데이터에 대한 가중치 별 중요도를 측정하고 이를 사용하여 새로운 데이터를 학습 시 과거 학습데이터에게 중요도가 높은 가중치의 변화를 억제하는 방법으로 LwF와 비교하여 직접적으로 과거 학습 데이터의 지식을 보존한다. Synaptic Intelligence(SI)[3,7] 방법은 학습 과정에서 매 배치(Batch)마다 변화하는 손실값의 변화량을 측정하고 이

를 최소화 하는 방법으로 과거 학습 데이터의 지식을 보존한다.

이러한 방법은 과거 학습 데이터를 저장하는 추가 메모리가 필요 없고 모델의 변동이 없거나 적은 특징을 갖고 있다. 또한 유사한 데이터들을 순차적으로 학습할 때 좋은 성능을 보인다[5].

2.2.2. 네트워크 동적 확장 방법

네트워크 동적 확장 방법은 필요에 따라 모델의 구조를 변경하여 모델의 정보 저장 능력을 동적으로 설정하는 방법이다. Dynamically Expandable Networks(DEN) [8] 방법은 희소 정규화를 사용하여 모델을 훈련하고 각 레이어에 추가할 뉴런 수를 결정하는 선택적 재훈련을 수행하여 학습한다. 이러한 방법은 학습하는 데이터의 특징에 따라 모델의 크기가 동적으로 커지는 특징을 가지고 있다.

2.2.3. 메모리 재생 방법

메모리 재생 방법은 과거 학습 데이터를 생성하거나 저장하는 방법을 사용하여 새로운 학습 데이터를 학습할 때 생성(저장)된 데이터를 추가해 학습하여 이전 학습 데이터의 정보를 유지하는 방법이다. Deep generative replay[9] 방법은 과거 학습 데이터를 생성하는 생성 모델을 추가로 학습하여 사용한다. 이러한 방법은 과거 학습 데이터를 저장하는 공간이 필요하거나 생성 모델을 위한 공간이 필요하므로 추가 메모리가 필요하고 생성 모델을 사용하는 방법인 경우 2가지 모델을 동시에 학습해야 하므로 학습 과정이 복잡한 특징을 가지고 있다.

2.3. 본 논문의 적응적 가중치 제어 방법

LwF는 모델의 출력값을 유지하려는 방법을 통해 이전 태스크의 정보를 유지하는 방법을 사용한다. 이 방법은 과거 학습 데이터를 사용할 필요 없이 현재 학습 데이터만 필요로 하는 장점을 가지고 있다. 과거 학습 데이터를 저장하지 않는 방법들 중 LwF는 비교적 높은 성능을 보인다고 연구되었다[5]. LwF는 CNN 레이어와 FC 레이어로 이루어진 기본적인 신경망 구조에 태스크마다 독립적인 Softmax 레이어가 존재하는 구조이다. 매 태스크마다 독립적인 Softmax 레이어가 추가되므로 증가되는 매개 변수의 수는 새 클래스의 수에 마지막 공유 계층의 노드 수를 곱한 값과 같다. 일반적으로 추가되는 매개 변수의 수는 네트워크의 모든 매개 변수의 수

에서 적은 비율을 차지한다.

LwF는 이전에 학습한 데이터와 현재 학습할 데이터가 유사할수록 더 좋은 성능을 보이지만 유사하지 않은 경우에는 성능이 떨어지는 모습을 보인다[5]. 성능 하락을 최소화하기 위해 태스크 출력 비중을 조절하는 가중치 λ 값을 상황에 맞게 선택하여 학습하는 방법을 사용할 수 있지만, LwF에서는 명확한 기준을 제시하지 않고 있으며, LwF에서는 데이터에 따라 0.15~2.5 또는 0.005~0.25 과 같은 적절한 λ 값을 상수값으로 선택해 사용하고 있다. 이 방법은 각 데이터 셋에 맞는 실험을 반복하여 최적의 λ 값을 찾아낸 뒤 최상의 값을 최종적으로 적용하는 방법이기 때문에 효율적인 방법이라고 하기 어렵다. 따라서 학습할 데이터의 특징에 따라 어떤 λ 값이 적절한지 판단할 수 있는 기준이 필요하다. 따라서 본 논문에서는 최적의 λ 값을 적응적으로 선택하여 적용하도록 방법을 제안하며, 개선된 성능을 얻을 수 있도록 λ 값을 설정하는 방법도 제안한다. 다양한 데이터가 들어오는 CL 시나리오를 구성하고 LwF를 사용하여 테스트를 수행하여, 데이터셋의 특성에 따른 최적의 λ 범위를 찾아내고, 이를 적응적으로 적용할 수 있는 알고리즘을 제안한다.

2.3.1. 사전 실험

2.3.1.1. 사전 실험 환경

Table. 1 Task scenario setting

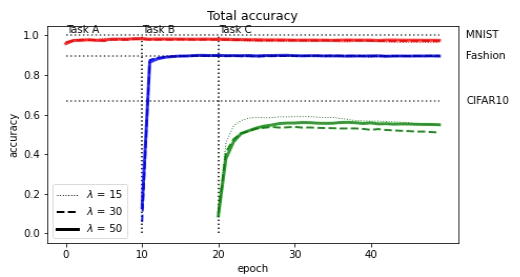
Task Scenario	Task A	Task B	Task C
1	Mnist	Fashion	Cifar10
2	Cifar10	Fashion	Mnist
3	Mnist	Emnist(Eng)	Cifar10
4	Cifar10	Emnist(Up)	Mnist
5	Fashion	Emnist(Eng)	Cifar10
6	Cifar10	Emnist(Eng)	Fashion

본 논문에서는, 표 1과 같이 실험 시나리오를 구성하여 사용한다. 이때 사용한 데이터는 총 5가지로 Mnist, Emnist(Eng), Emnist(Up), Fashion Mnist, Cifar10으로 구성되어 있다. Emnist 데이터 셋 중 영문자를 모두 포함한 데이터셋은 Emnist(Eng), 영문 대문자만 포함한 데이터셋은 Emnist(Up)로 구분한다. 시나리오를 구성할 때 학습 후 최고 정확도에 따라 데이터의 학습 난이도를 구분하였는데 학습 난이도가 낮은 데이터로는

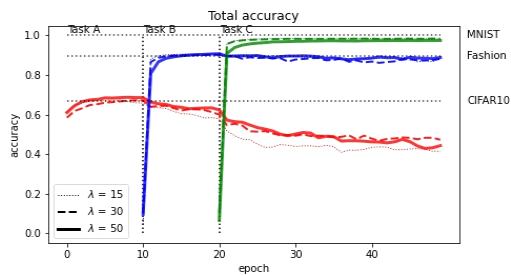
Mnist, Emnist(Up) 데이터를 할당하였고 학습 난이도가 높은 데이터로는 Fashion Mnist, Emnist(Eng), Cifar10 데이터를 할당하였다.

시나리오 1, 3, 5는 상대적으로 학습하기 간단한 데이터셋으로 시작해 점차 학습하기 복잡한 데이터셋이 들어오는 시나리오, 시나리오 2, 4, 6은 상대적으로 학습하기 어려운 데이터셋으로 시작해 점차 학습하기 쉬운 데이터셋이 들어오는 시나리오로 구성하였다. 학습 배치 크기는 100개로 설정하였고 에포크(epoch)는 각 태스크 별로 10회 (마지막 태스크는 30회)로 설정, λ 값은 15.0, 30.0, 50.0 으로 임의로 설정하여 비교를 수행하였다. 대부분의 LwF 관련 논문에서는 1에서 50 사이의 값을 상수로 설정하여 진행하고 있다.

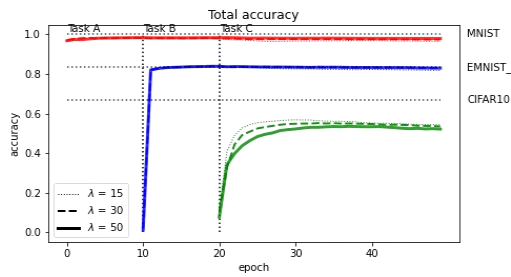
2.3.1.2. 사전 실험 결과



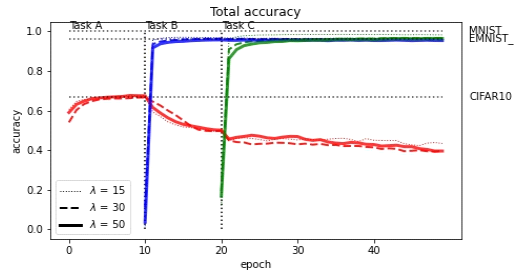
(a) scenario 1



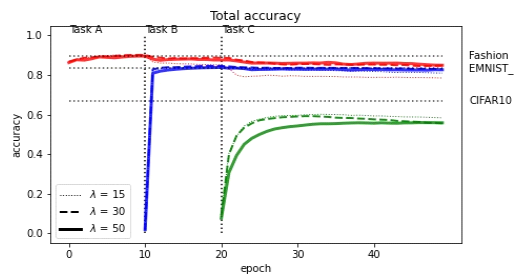
(b) scenario 2



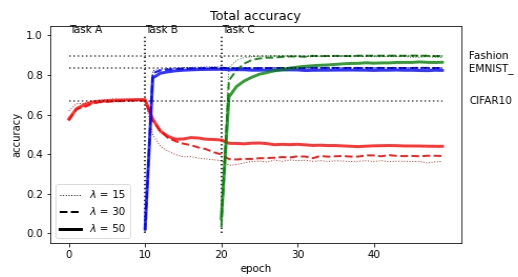
(c) scenario 3



(d) scenario 4



(e) scenario 5



(f) scenario 6

Fig. 1 Accuracy of each scenario

Table. 2 Accuracy of each scenario at 30 epoch

Senario/lambda	Task	Task			
		A	B	C	Avg
1	15.0	97.0	89.8	58.7	81.8
	30.0	97.5	89.6	53.4	80.2
	50.0	97.7	89.9	55.6	81.1
2	15.0	44.4	89.6	98.3	77.4
	30.0	49.5	86.7	98.3	78.2
	50.0	49.2	89.5	96.8	78.5
3	15.0	96.6	82.6	56.6	78.6
	30.0	97.6	83.5	54.6	78.6
	50.0	98.0	83.4	51.8	77.7
4	15.0	45.5	96.4	98.4	80.1
	30.0	43.4	96.1	95.8	78.4
	50.0	46.9	95.6	95.3	79.3

Task		A	B	C	Avg
Senario/lambda					
5	15.0	80.1	82.8	60.2	74.3
	30.0	85.2	83.4	58.7	75.8
	50.0	85.9	92.9	53.8	74.2
6	15.0	37.0	83.6	89.5	70.0
	30.0	38.3	83.5	89.3	70.4
	50.0	44.7	82.5	83.4	70.2

그림 1은 학습이 진행됨에 따라 각 태스크들의 정확도 변화량을 람다(λ)값에 따라 그래프로 나타낸 것이다.(가로 점선은 각 시나리오에서 사용한 데이터 셋을 단독으로 학습 했을 때의 최대 정확도 이다) 논문에서 대상으로 하는 각 시나리오에 대해 태스크가 진행됨에 따라 어느정도 망각이 일어나는지를 람다값을 달리하며 나타낸 것이다. 특정 상황에 맞는 람다값은 해당 실험에 최적으로 설정하여 사용해도 되겠지만, 본 논문의 대상인 지속적 학습환경 또는 온라인 학습환경에서는 매 태스크마다 데이터 특성이 달라질 수 있으므로, 그림 1에서 보는 것과 같이 람다값에 따라 결과 차이가 커질 수 있는 것이다. 표 2는 전체 학습 에포크 수가 30 일 때, 즉 각 태스크 별 10 에포크 씩 학습했을 때의 정확도를 나타낸 표이다. 시나리오에 따라 람다값이 망각률에 영향을 미치는 것을 확인할 수 있다.

2.2.1.3. 복잡도(학습 난이도)에 따른 관점

그림 1의 (a) 그림은 태스크가 진행될수록 복잡한(학습 난이도가 높은) 데이터가 들어오는 시나리오의 정확도이다. 이 시나리오는 Task B, C가 학습됨에 따라 다른 태스크들의 정확도가 하락하는 망각 현상은 보이지 않는다. (b) 그림은 태스크가 진행될수록 단순한(학습 난이도가 낮은) 데이터가 들어오는 시나리오의 정확도이다. 태스크가 진행될수록 복잡한 데이터인 Task A의 정확도가 떨어지는 모습을 보인다. 또한 (d), (e), (f) 그림에서 동일하게 복잡한 데이터인 Task A의 정확도 하락이 관찰되었다. 치명적 망각 현상은 데이터가 복잡할수록(학습 난이도가 높을수록) 더 잘 일어난다.

2.3.1.4. 데이터 지속에 따른 관점

그림 1 Task C의 에포크 수는 Task A, B 보다 3배 더 많다. (e) 그림의 Task C 정확도를 보면 λ 값이 높을수록 최고 정확도에 도달하는 에포크가 커진다. 따라서 λ 가 높을수록 학습 에포크수를 늘려야 한다. 하지만 (b), (d)

그림의 경우 Task C의 에포크 수가 10회인 경우와 30회인 경우를 비교하면 Task B, C의 정확도는 유지되는 반면 Task A의 정확도는 낮아지는 모습을 보인다. 따라서 높은 λ 값을 설정하고 단순히 에포크 수를 늘리는 것은 좋지 않게 된다.

2.3.1.5. 정규화 강도에 따른 관점

그림 1의 (e)를 살펴보면 λ 값이 커질수록 이전에 학습한 Task A에 대한 정확도는 상대적으로 더 유지되는 반면 새롭게 학습하는 Task C에 대한 정확도는 낮아지는 것을 볼 수 있다. 이 현상은 (c), (f) 에서도 볼 수 있는데 새롭게 학습하는 Task C의 정확도의 상승폭이 낮은 λ 값 보다 적어진 것을 알 수 있다. 이전 학습 데이터의 정보를 유지하기 위해 무조건 λ 값을 크게 설정하는 방법은 좋지 않다.

2.3.2. 가중치 할당 방안

LwF 에서는 이미 학습된 데이터와 새로 학습할 데이터의 유사도가 떨어지면 LwF의 성능이 떨어진다고 하였다[4]. 그림 1과 표 2에서 볼 수 있듯이 다른 학습 데이터와 유사도 차이가 큰 복잡한 데이터(3채널 컬러 이미지)는 다른 데이터들과 다르게 영향을 크게 받는 것을 알 수 있다. 이러한 영향을 최소화하기 위해서는 λ 값을 적절하게 조정하여 새로 학습한 데이터와 이전에 학습한 데이터의 정확도 손실을 최소화 하는 것이 중요하다. 실험결과에서도 알 수 있듯이, λ 값이 커질수록 이전에 학습한 태스크의 정확도는 유지되는 반면 새로 학습할 태스크의 정확도는 떨어지는 모습을 보이기 때문에 λ 값을 적응적으로 설정하는 것이 매우 중요하다.

이를 위해 본 논문에서는, 신경망의 활성화 출력을 이용해 유사도를 측정하고 λ 값을 정하는 방법을 사용한다. 먼저 두 데이터 사이의 유사도를 측정하기 위해 신경망의 활성화 출력값을 이용한다[10]. 두 데이터 사이의 활성화 출력값이 비슷하다면 학습과정에서 서로 영향을 많이 주게되고, 그렇지 않다면 영향이 적어지게 된다.

이를 이용하여 상관계수와 λ 값을 단순하게 연동시키면, 상관계수가 높은 경우, λ 값을 상대적으로 높여 이전에 학습한 태스크에 중점을 두어 학습하게 하고, 상관계수가 낮다면 λ 값을 낮추어 현재 학습할 태스크에 중점을 두어 학습하는 방법을 생각할 수 있다. 하지만 이 방법을 사용했을 경우 문제가 발생한다.

Table. 3 Output Distribution Correlation Coefficient Between Learning Data

New Old	Mnist	Emnist (Eng)	Emnist (Up)	Fashion Mnist	Cifar10
Mnist	-	0.829	0.821	0.645	0.537
Emnist (Eng)	0.857	-	0.977	0.833	0.663
Emnist (Up)	0.833	0.978	-	0.804	0.617
Fashion Mnist	0.653	0.729	0.761	-	0.679
Cifar10	0.357	0.567	0.522	0.516	-

표 3은 학습된 태스크 데이터와 새로운 태스크 사이의 출력 분포 상관계수를 측정한 결과이다. Cifar10과 Mnist의 상관계수는 다른 데이터와 비교했을 때 0.357-0.537로 낮게 측정된다. 이때 상관계수가 낮으므로 λ 값을 큰 값으로 설정하게 된다면 그림 2의 (f)과 같이 유사도가 낮고 복잡한 데이터인 이전에 학습한 Cifar10의 성능을 유지하는 모습을 보일 것이다. 반면에 그림 2의 (e)는 새롭게 학습한 Cifar10의 성능이 떨어지는 결과를 보이는데 이는 같은 데이터라도 학습 순서, 즉 유사도가 낮고 복잡한 데이터가 이전에 학습한 태스크인지 현재 학습할 태스크 인지 따라 λ 값을 다르게 설정해야 한다는 것을 알 수 있다.

일반적으로 동일하거나 유사한 특징을 갖는 데이터가 들어오는 Incremental Domain Learning 환경은 학습 순서의 중요도가 상대적으로 떨어지지만, LwF가 속한 Incremental Task Learning 환경은 서로 다른 특징을 가진 데이터들이 들어오는 환경이므로, 학습 순서는 고려해야 할 사항 중 하나이다. 따라서 유사도뿐만 아니라 데이터의 복잡도 또한 측정할 수 있어야 망각현상을 최대로 제어할 수 있게 된다.

2.3.3. 복잡도 측정 방법

복잡도를 측정하기 위해 본 논문에서는, LwF에서 이용한 워업(warm-up) 방법을 사용한다. 워업은 데이터를 학습하기 전 공유 레이어는 고정시키고, 태스크 의존 레이어, 즉, softmax layer만 학습하는 방법이다. LwF에서는, 이 방법이 LwF에서 성능향상의 효과가 없기 때문에 필수적이지 않다고 하였다[4]. 본 논문에서는 워업방법을 직접적인 성능향상을 위한 목표가 아니라, 정확도 측정을 위한 방법으로 사용한다. 즉, 새로운 데이터로 워

업 과정을 거치고 정확도를 측정한 후 이전 태스크의 정확도와 비교하여 현저하게 낮게 측정되면 복잡한 데이터, 비슷하거나 약간 낮게 측정되면 단순한 데이터로 판별하는 방법을 사용하는 것이다. 워업 과정을 거쳐 측정된 정확도는 공유 레이어가 고정되어 학습되었기 때문에 이전 태스크보다 낮게 나오는 경향이 있으므로 이를 감안하여 기준을 정해야 한다. 따라서 워업 과정을 거친 현재 태스크의 정확도가 이전 태스크 정확도의 절반보다 낮게 측정되면 복잡한 데이터로 판별하게 하였다. 표 4에 본 논문에서 제안하는 학습 과정과, 측정한 유사도와 복잡도를 사용하여 λ 를 설정하는 방법을 알고리즘으로 나타내었다.

Table. 4 Training Algorithm

Algorithm 1. Training Algorithm
Require: Dataset, $D = (D_1, \dots, D_T)$, α, β for each Task $t = 1, \dots, T$ do if $t = 1$ then Train the network using normal cross entropy loss Calculate Task 1 accuracy acc_1 Calculation network activation value A_1 else Calculate network activation value A_t Make output layer for Task t Do warm-up process Calculate Task t accuracy acc_t Calculate λ using Algorithm 2 Make output value for old Tasks using the network and D_t Train the network using LwF method Update A_t, acc_t
Algorithm 2. Lambda Calculation
Require: network activation value A_{t-1}, A_t Task accuracy acc_{t-1}, acc_t α, β Calculate Correlation Coefficient x with A_{t-1}, A_t [10] if $(acc_{t-1} \times 0.5) < acc_t$ then return $-\ln(-x + 1) \times \alpha$ else return $-\ln(x) \times \beta$

표 4의 Algorithm 1은 제안하는 학습 과정을 나타낸 알고리즘이다. 기본적으로 LwF 방법을 기반으로 수행되며 Task 2 이상부터 제안하는 방식을 적용하여 학습을 수행한다. 새로운 태스크를 학습하기 전 상관계수 측정을 위한 전 네트워크의 출력값 A_t 을 측정하고 워업과정을 수행하여, 복잡도 측정을 위한 현재 태스크의 정확

도 acc_t 를 측정한다.

이후 Algorithm 2를 사용하여 적합한 λ 값을 측정하여 LwF 학습에 사용한다. 표 4의 Algorithm 2는 본 논문의 λ 계산 방법을 나타낸 알고리즘이다. x 는 상관계수이고 0~1 사이의 값을 갖는다. 만약 현재 태스크의 정확도 acc_t 가 이전 태스크의 정확도 acc_{t-1} 의 절반보다 높게 나온다면 유사도가 높고 단순한 데이터로 판별하여 λ 를 계산하게 되고, 아닐 경우 유사도가 낮고 복잡한 데이터로 판별하여 λ 를 계산한다. 전자의 λ 는 0~ ∞ 사이의 값을 갖고(x 값이 1에 접근할수록 λ 는 ∞ 에 수렴하는 형태). 후자의 λ 는 ∞ ~0사이의 값을 갖는다(x 값이 0에 접근할수록 λ 는 ∞ 에 수렴하는 형태). α, β 는 얼마나 값이 급격하게 변하는지 설정하는 값이다. 이 실험에서 α 는 30, β 는 100로 설정하였다.

2.4. 본 논문의 실험 결과

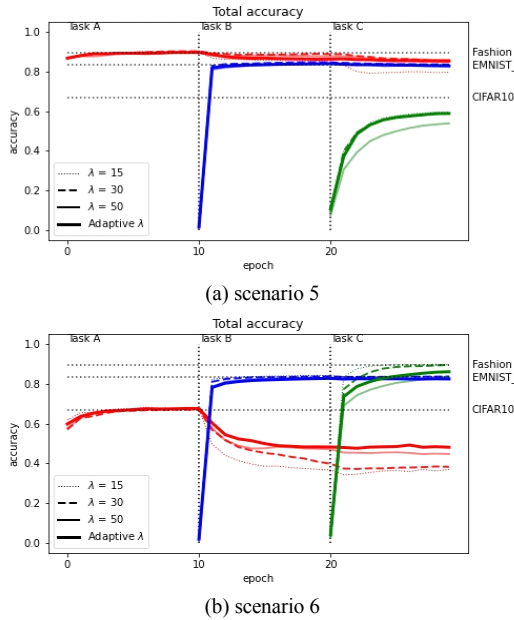


Fig. 2 Accuracy of scenario 5, 6 with adaptive lambda

Table. 5 Accuracy of each scenario with adaptive lambda

Senario	Task	A	B	C	Avg
1		97.5	89.1	54.3	80.3
2		51.5	89.2	96.9	79.2
3		97.3	83.1	53.6	78.0

Senario	Task	A	B	C	Avg
4		47.0	94.9	98.0	80.0
5		85.1	82.3	58.8	75.5
6		48.1	82.3	85.9	72.1

본 논문의 방법에 의한 실험결과 중, 정확도 개선의 관점의 결과를 그림 2와 표 5에 나타내었다. 이것은 가변 λ 방법과 기존의 고정된 λ 값을 가진 LwF 방법을 비교한 결과를 나타낸 것이다. 그림 2의 (a)와 표 5의 시나리오 5에서 Task C의 경우, 50의 고정된 값 보다 가변 λ 방법의 정확도가 약 5% 높은 것을 알 수 있다. 그리고 15의 고정된 λ 값 보다 Task A의 정확도가 약 5% 높게 측정된다. 또한 (b)의 Task A를 보면 15의 고정된 값 보다 가변 λ 방법의 정확도가 약 11% 상승, 50의 고정된 값 보다 약 4% 높은 값이 나오는 모습을 보여준다. 이러한 결과들을 통해, 본 논문의 방법으로 λ 를 고정된 낮은 값이나 높은 값으로 설정하는 것 보다, 평균적으로 더 효율적인 결과를 얻을 수 있다는 것을 알 수 있다. 또한 LwF의 학습 환경은 일관된 특징을 가진 데이터들이 들어오는 환경이 아닌 다양한 특징을 가진 데이터들이 들어올 수 있는 Incremental Task Learning 환경이다. 이 환경에서 중요한 고려사항인 학습 순서를 고려한 가변 λ 방법을 적용함으로써 그림 2의 그래프와 같이 정반대의 학습순서로 학습을 수행하더라도 본 논문의 방법은 좋은 결과를 얻을 수 있었다.

표 6에 각 태스크를 수행할 때마다, 실험에 의해 수동으로 찾아낸 최적의 람다값(Best λ)과 본 논문에서 제안한 방법으로 자동으로 설정된 람다값(Adaptive λ)을 비교하여 나타내었다. 수동으로 최적의 람다값을 설정하기 위해 새로운 태스크 학습 시 임의의 여러 람다값들을 (15, 20, 30, 40, 50) 모두 적용시켜 실험하였고, 그 중 평균 정확도가 높게 측정되는 람다값을 선택하는 방법을 사용하였다.

Table. 6 Comparison Between Best Lambda and Adaptive Lambda

Senario/lambda	Task	A->B	B->C
	1	Best	30
	Adaptive λ	42	36
2	Best	50	30
	Adaptive λ	67	50

Senario/lambda	Task	A->B	B->C
3	Best	20	20
	Adaptive λ	19	26
4	Best	50	15
	Adaptive λ	78	20
5	Best	50	20
	Adaptive λ	26	22
6	Best	50	15
	Adaptive λ	93	34

시나리오 1, 3, 5의 경우 점차 학습하기 어려운 데이터셋이 들어오는 시나리오, 시나리오 2, 4, 6은 점차 학습하기 쉬운 데이터셋이 들어오는 시나리오이다. 제안하는 방법을 사용한 결과에서 Task A 학습 후 Task B를 학습할 때 시나리오 1, 3, 5는 각각 2, 4, 6 시나리오에 비해 비교적 낮은 램다 값으로 시작하는 경향이 보이고 수동으로 측정된 램다 값도 이와 같은 경향을 보인다.

이는 상대적으로 학습이 어려운 데이터들을 학습할 때 학습 성능 향상을 높여주는 역할을 하므로 최적 값으로 볼 수 있다. 반대로 시나리오 2, 4, 6의 경우 램다 값이 높게 설정되므로 과거 학습 데이터들의 성능 유지에 효과적이다. 따라서 이 값 또한 최적 값으로 볼 수 있다.

Task B학습 후 Task C를 학습할 때 시나리오 1, 3, 5의 경우 어려운 데이터 셋(Cifar10)의 원활한 학습을 위해 모든 방법에서 상대적으로 낮은 램다값이(30대 이하) 선택되어 최적 값으로 설정된 것을 볼 수 있다. 시나리오 2, 4, 6의 경우 시나리오 2를 제외한 모든 경우에서 상대적으로 낮은 램다 값이 설정된 것을 볼 수 있다. 이는 표 2에서도 알 수 있듯이, 시나리오 4(Emnist - mnist), 시나리오 6(Fashion - Emnist)의 데이터 사이의 유사도가 시나리오 2(Fashion - Mnist) 사이의 유사도보다 크기 때문에 나타나는 현상으로, 유사도가 크고 복잡도가 낮은 데이터가 들어오는 경우 램다값이 낮은 값으로 설정되지만 시나리오 2와 같이 유사도가 낮은 경우 램다값이 높은 값으로 설정되어야 하기 때문이다. 따라서 이 값들 또한 최적 값으로 볼 수 있다.

대부분의 시나리오에서 본 논문에서 제안한 방법은, 수동으로 선택한 최적의 램다 값과 유사한 변화 양상을 보이고 있다. 상관 계수를 계산한다면 약 0.739의 값으로 측정된다. 이 결과들을 통해 제안하는 방법은 새로운 태스크를 학습할 때마다 적절한 램다 값을 설정하였으

며, 여러 가지 시나리오에서 최적의 결과값을 도출하고 있다는 것을 알 수 있다.

III. 결론

LwF 방법은 다양한 종류의 데이터를 차례로 습득하여 학습하는 지속적 학습(CL)환경에서 발생하는 치명적인 망각 현상을 억제할 수 있는 효과적인 방법 중 하나이다. LwF는 이전 태스크 데이터의 정보를 얼마나 유지할 것인지 설정하는 값인 λ 가 존재하는데 이 값에 따라 현재 학습할 태스크와 이전에 학습했던 태스크 사이의 상관관계를 설정할 수 있다. 만약 λ 값을 낮게 설정하게 된다면 이전 태스크의 망각 현상이 커지게 되고, 이전 태스크의 망각 현상을 최소화하기 위해 단순히 λ 값을 큰 값으로 설정할 경우 현재 태스크의 학습에 제약이 생기게 되어, λ 값이 낮은 경우보다 정확도가 낮아지게 되고 학습 시간이 늘어나게 되는 문제점이 있다. 따라서 본 논문에서는 이전 태스크와 새로운 태스크의 특징과들 사이의 관계를 파악하여, LwF 기반의 학습모델에 맞는 λ 값을 적응적으로 설정하는 방법을 제안하였다. 그 결과 새로운 태스크의 정확도가 최대 5%, 이전 태스크의 정확도가 최대 11% 상승하였다. 이러한 결과를 통해 단순히 고정된 λ 값을 선택하는 게 아닌 학습 데이터의 특성을 파악하여 가변적으로 λ 값을 선택하는 방법이, 낮은 λ 값과 높은 λ 값 사이의 효율적인 상관관계값을 제시하며, 치명적인 망각 현상을 억제하면서 다양한 특징을 가진 데이터들을 효과적으로 학습할 수 있음을 나타낸다. 향후에는 더 복잡하고 큰 모델 및 더 지속되는 태스크 처리 환경에서의 망각현상 개선을 검증하고 필요한 개선 방법을 개발할 필요가 있다.

REFERENCES

- [1] I. J. Goodfellow, M. Miraza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211*, 2013.
- [2] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp.

- 128-135, Apr. 1999.
- [3] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54-71, 2019.
- [4] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935-2947, Nov. 2017.
- [5] Y. Hsu, Y. Liu, A. Ramasamy, and Z. Kira, "Re-evaluating continual learning scenarios: A categorization and case for strong baselines," *arXiv preprint, arXiv:1810.12488*, 2018.
- [6] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521-3526, 2017.
- [7] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 3987-3995, 2017.
- [8] J. Yoon, E. Yang, J. Lee, and S. Hwang, "Lifelong learning with dynamically expandable networks," *arXiv preprint arXiv:1708.01547*, 2017.
- [9] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *arXiv preprint arXiv:1705.08690*, 2017.
- [10] S. H. park and S. H. Kang, "Continual Learning using Data Similarity," *Institute of Korean Electrical and Electronics Engineers*, vol. 24, no. 2, pp. 514-522, 2020.



박성현(Seong-Hyeon Park)

2020 3월~ 현재, 임베디드 시스템 공학과 석사과정
 ※관심분야: 인공지능, 딥러닝, 임베디드 시스템



강석훈(Seok-Hoon Kang)

1995년 8월 한양대학교 전자통신공학과 공학박사
 2004년 3월 ~ 현재 인천대학교 임베디드시스템공학과 교수
 ※관심분야: 인공지능, 딥러닝, 인식시스템, 임베디드 시스템, 모바일 시스템