한국컴퓨터그래픽스학회
Korea Computer Graphics Society Vol. 29, No. 3, P. 85~92

누적 가중치 변화의 시각화를 통한 심층 신경망 분석시스템

양태린O, 박진호
숭실대학교 미디어학과
ytltg146@soongsil.ac.krO, c2alpha@ssu.ac.kr

Deep Neural Network Analysis System by Visualizing Accumulated Weight Changes

Taelin YangO, Jinho Park
Department of Digital Media, Soongsil University

요 약

최근 ChatGPT나 자율주행 자동차 등의 인공지능 분야의 급속한 발전으로 인해 인공지능에 대한 관심이 높아졌다. 그러나 아직 인공지능은 학습 과정에서 알 수 없는 요소가 많이 존재하여 모델을 개선하거나 최적화하기 위해서 필요 이상의 시간과 노력을 들여야 하는 경우가 많다. 따라서, 인공지능 모델의 학습 과정에서 가중치 변화를 명확하게 이해하고 해당 변화를 효과적으로 분석할 수 있는 도구 또는 방법론이 절실하게 요구되고 있다. 본 연구에서는 이러한 점을 해결하기 위해 누적 가중치 변화량을 시각화해주는 시스템을 제안한다. 시스템은 학습의 일정한 기간마다 가중치를 구하고 가중치의 변화를 누적시켜서 누적 가중치로 저장하여 3차원 공간상에 나타내게 된다. 이로 인해 보는 이로 하여금 한눈에 레이어의 구조와 현재의 가중치 변화량이 이해되기 쉽게 구성하였다.
이러한 연구를 통해 인공지능 모델의 학습 과정이 어떻게 진행되는지에 대한 이해와 모델의 성능 향상에 도움이 되는 방향으로 하이퍼 파라미터를 변경할 수 있는 지표를 얻게 되는 등 인공지능 학습 과정의 다양한 측면을 탐구할 수 있을 것이다. 이러한 시도를 통해 아직 미지의 영역으로 여겨지는 인공지능 학습 과정의 일부를 보다 효과적으로 탐색하고 인공지능 모델의 발전과 적용에 기여할 수 있을 것으로 기대된다.

Abstract

Recently, interest in artificial intelligence has increased due to the development of artificial intelligence fields such as ChatGPT and self-driving cars. However, there are still many unknown elements in training process of artificial intelligence, so that optimizing the model requires more time and effort than it needs. Therefore, there is a need for a tool or methodology that can analyze the weight changes during the training process of artificial intelligence and help out understatnding those changes. In this research, I propose a visualization system which helps people to understand the accumulated weight changes. The system calculates the weights for each training period to accumulates weight changes and stores accumulated weight changes to plot them in 3D space.
This research will allow us to explore different aspect of artificial intelligence learning process, such as understanding how the model get trained and providing us an indicator on which hyperparameters should be changed for better performance. These attempts are expected to explore better in artificial intelligence learning process that is still considered as unknown and contribute to the development and application of artificial intelligence models.

키워드: XAI, 시각화, 하이퍼 파라미터 최적화, 누적 가중치
Keywords: XAI, Visualization, Hyper-parameter optimization, Accumulated Weight

*corresponding author: Jinho Park / Department of Digital Media, Soongsil University (c2alpha@ssu.ac.kr)

Received : 2023.06.16./ Review completed : 1st 2023.06.30. / Accepted : 2023.07.05.
DOI : 10.15701/kcgs.2023.29.3.85
ISSN : 1975-7883(Print)/2383-529X(Online)

1. 서론

최근 OpenAI에서 개발한 ChatGPT나 많은 자동차 회사에서 관심을 가지는 자율주행 자동차 등의 인공지능 분야의 급속한 발전으로 사람들의 관심도 크게 증가하고 있다. 하지만, 인공지능 모델의 복잡성과 아직은 부족한 해석력으로 인해 학습 과정에서 발생하는 가중치 변화와 이에 관련된 여러 요소들에 대한 정확한 이해와 설명이 부족한 실정이다[1].

인공지능 알고리즘, 특히 딥러닝 모델의 학습 과정에서 가중치 변화와 관련하여 다양한 요소들이 존재하지만, 아직 원인과 결과 간의 관계가 어떻게 형성되는지, 모델의 레이어 개수와 각 레이어의 가중치 개수가 모델 최적화와 어떠한 연관이 있는지 등 명확한 기준이 부족한 상태이다. 이로 인해 현재는 정해진 방법보다는 무작위로 설정하고 결과에 따라 조금씩 변경해 보는 방법을 대부분 사용하기에 학습 과정에서 어떤 요소가 성능에 영향을 미치는지 명확하게 파악하기 어렵다[2].

이러한 원인들로 인해, 인공지능 연구에 있어서 모델을 개선하거나 최적화하기 위해서 필요 이상의 시간과 노력을 들여야 하는 경우가 많다. 따라서, 인공지능 모델의 학습 과정에서 가중치 변화를 명확하게 이해하고 해당 변화를 효과적으로 분석할 수 있는 도구 또는 방법론이 절실하게 요구되고 있다[3].

본 논문에서는 인공지능 학습 과정에서 발생하는 가중치를 학습이 진행됨에 따라 점차 변화하는 정도를 측정하여 그 누적값을 시각화하는 프로그램을 개발하고, 이를 3차원 공간상에 표시하여 인공지능 모델의 성능과 학습 정확도에 미치는 영향을 분석한다. 이러한 연구를 통해 모델의 레이어 개수와 각 레이어의 가중치 개수와 관련하여 좀 더 명확한 지침과 이해를 제공하여, 연구자들이 최적화된 인공지능 모델을 개발하는 일에 도움이 될 것으로 기대된다. 또한 이러한 시도를 통해 블랙박스로 묘사되는 인공지능 학습 과정의 일부를 해결하고, 인공지능 모델의 발전에 조금 더 기여할 수 있기를 기대한다.

2. 관련 연구

2.1 설명가능한 인공지능

인공 지능 분야에서 데이터 시각화를 활용하여 인공지능의 학습 과정과 선택 이유 등을 이해하고자 시각적 분석을 시도하는 XAI(Explainable Artificial Intelligence)는 많은 시도가 있었다. 대표적인 예로 2015년에 Bolei Zhou 등의 연구자 들이 발표한 CAM(Class Activation Mapping) 은 기존의 CNN (Convolutional Neural Network) 기반 인공지능 모델의 FC-Layer를 대체하기 위한 프레임워크로, Global Average Pooling (GAP)을 적용하여 CNN이 이미지에서 object localization을 학습할 수 있도록 만들었다[4].

CAM을 이용하면 다양한 클래스들의 특성을 고려한 heatmap을 생성할 수 있고, 이를 통해 네트워크가 이미지의 어느 부분을 참조하여 판단했는지에 대한 분석이 가능하다. 주목할 점은 CAM이 지도학습 없이도 heatmap을 생성할 수 있다는 사실이다. 이를 통해 연구자들은 각 클래스들의 특정 특성을 잘 파악하고 추출해 낸 것으로 확인할 수 있다.

또 다른 사례로는 2016년에 Marco Tulio Ribeiro가 제시한 LIME 알고리즘이 있다. LIME 알고리즘은 다양한 모델에 적용할 수 있는 범용적인 알고리즘으로, 기본 아이디어는 모델의 복잡한 결정 과정을 지역적으로 근사화하여 이해하기 쉬운 모델로 설명할 수 있다는 것이다. 즉 전체 데이터에 대한 해석이 아니라 데이터마다 사람이 해석 가능한 표현으로 재구성하여 해석하는 국소적 대리 분석 방법을 수행하는 알고리즘이다[5].

이러한 방식은 다음과 같은 과정을 거친다

(1) 주어진 데이터의 지역적 변동성을 살펴봄으로써 모델의 예측 결과를 설명한다.

(2) 해당 데이터 포인트 주변에서 작동하는 설명 가능한 모델을 찾는다. 설명 가능한 모델은 선형 회귀나 결정 트리와 같이 해석하기 쉬운 구조를 가진다.

(3) 복잡한 기계학습 모델과 비교하여 설명 가능한 모델이 유사한 결과를 얻을 수 있도록 조정한다.

이것 외에도 XAI는 많은 시도가 이루어지고 있는 분야이다. 그러나 이러한 관심에도 불구하고 왜 그런 결과가 나왔는지에만 집중하고, 어떻게 그 결과가 나왔는지 그 과정에는 큰 관심을 주지 않는 경향이 있다. 따라서 본 논문에서는 가중치 변화량을 시간에 따라 시각화하여 단순하게 결과에 치중하지 않고 인공지능을 연구하는 사람들에게 도움을 줄 수 있는 도구를 제시한다.

2.2 하이퍼 파라미터 최적화

신경망 모델의 성능을 향상시키기 위해 고려해야 할 중요한 하이퍼 파라미터로 레이어의 개수, 각 레이어의 크기, 가중치 초깃값 등이 있다. 가중치 초깃값 설정에 관해서는 이미 다양한 연구로 정형화된 방법들이 존재한다. 예를 들어, Xavier 초기화와 He 초기화는 널리 사용되는 가중치 초기화 방법이다[6].

그러나 레이어의 개수와 크기에 관한 최적화 문제는 아직까지 확실한 정설이 없다. 따라서 현재까지도 이 부분에 대한 연구가 활발히 진행되고 있으며, 다양한 접근법이 시도되고 있다[7].

이러한 접근법 중 일부는 모델의 성능을 개선할 수 있는 신경망 구조를 설계하는 것을 목표로 한다.

예를 들면 최적화를 위해 그리드 탐색(Grid Search)[8] 과 베이지안 최적화(Bayesian Optimization)[9] 과 같은 기법을 말할 수

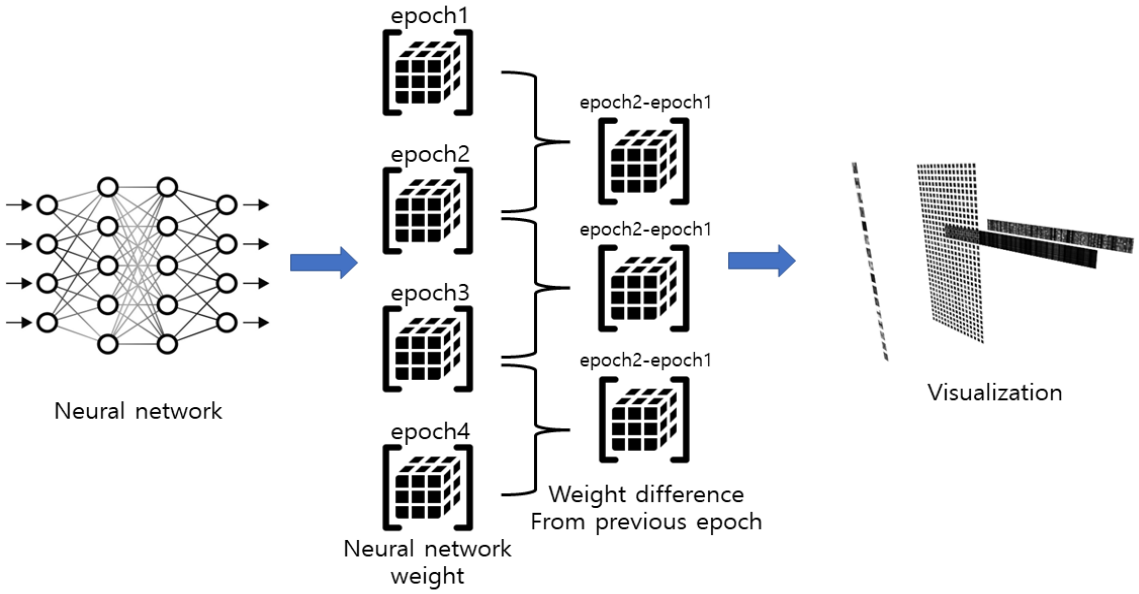

Figure 1: Overall schematic of visualizing system for accumulated weight

있는데, 먼저 그리드 탐색은 후보 하이퍼 파라미터 값들을 미리 정해두고, 해당 구간 내에서 특정 간격으로 탐색하여 각 하이퍼 파라미터 값들에 대한 성능을 측정하고 기록한다. 이후, 가장 높은 성능을 보인 하이퍼 파라미터 값을 선택하여 모델을 최종 조정하는 방법이다. 그리드 탐색은 교차 검증 결과가 가장 우수한 하이퍼 파라미터 값을 찾아내는 방법으로, 일일이 모든 가능한 조합을 시도하는 대신, 효율적으로 최적의 하이퍼 파라미터 값을 찾을 수 있는 방법이다.

베이시안 최적화 같은 경우에는 입력값을 받는 미지의 목적 함수를 상정하여, 해당 함숫값을 최대로 만드는 해를 찾는 것을 목적으로 대상 함수와 해당 함수의 하이퍼 파라미터 조합을 대상으로 Surrogate Model(대체 모델)을 만든다. 모델을 기반으로 순차적으로 하이퍼 파라미터를 업데이트하며 평가를 수행하여 최적의 하이퍼 파라미터 조합을 찾을 수 있다.

이러한 방법을 통해 다양한 하이퍼 파라미터 조합을 탐색하고, 최적의 구조를 찾아낼 수 있다. 따라서, 신경망 모델의 최적화에 대한 연구는 레이어의 개수와 크기에 관한 단순한 시간 단축 문제를 넘어서 기계학습 모델의 구조를 개선하는 방법을 제시하는 데 큰 기여를 하고 있다. 본 논문에서는 이러한 문제 해결을 위해 도움이 되는 하나의 도구를 제시하여 이 분야의 연구가 계속 진행됨에 따라, 더 높은 성능의 모델 개발에 기여할 것으로 기대된다.

3. 시스템 설계 및 분석

인공지능에서 네트워크를 시각화하는 방법에는 다양한 접근법이 있으나, 본 논문에서는 가중치 변화를 시각화하여 네트워크의 이해를 돕고자 한다. 전체적인 구성은 PyTorch를 사용하여 네트워크를 구성하고 네트워크에서 나온 가중치 값을 Unity와 연동하여 가중치를 시각화하는 시스템을 설계하였다. 본 장에서는 시스템 설계에 고려한 요소들과 이를 바탕으로 한 시스템 구조 및 방법론에 대해서 설명한다

3.1 시스템 설계

본 연구에서는, 가중치의 변화를 보고 그것을 토대로 다음 학습의 영향을 주기 위해 기존의 많은 논문에서 하던 가중치 시각화와는 조금 다르게 각 epoch마다 가중치를 저장했고, 이전 epoch과의 가중치 차이를 구하여 그 차이의 절댓값을 따로 저장했다. 이를 통해 학습이 진행됨에 따라 변하는 각 레이어의 전 레이어와의 가중치 차이를 구할 수 있다. 이렇게 가중치 차이를 확인하면 학습이 진행됨에 따라 매번 각 epoch마다 전 epoch에 비해 가중치가 얼마나 바뀌었는지를 절대적 수치로 알 수 있고 이를 다른 레이어와 비교하며 현재 학습이 진행됨에 따라 가장 많이 변하고 있는 레이어의 값을 확인할 수 있다. 따라서 현재의 학습에 가장 영향을 많이 끼치고 있는 레이어와 그렇지 않은 레이어를 확인 가능했다. (Figure 1 참조)

또한 각 레이어의 모습을 특징에 따라 다르게 저장하여 조금 더 가시성을 높이고 단순하게 이미지만 보고도 어떤 레이어의 모습인지 쉽게 알 수 있도록 구성했다.

시각화를 직접적으로 사용자에게 보여주는 플랫폼으로는 Unity를 활용하여 누적 가중치를 3차원으로 시각화하는 환경을 제공했다. 기존의 2D에서 보는 시각화에서는 한눈에 파악하기 어려운 레이어 구조와 변화를 보다 명확하게 이해할 수 있도록 배치하고 시간의 흐름에 따른 변화를 확인할 수 있는 방법을 도입하였다. 이를 통해 더욱 효과적인 가중치 변화 분석이 가능하게 되었다.

3.2 학습 네트워크 환경 설정

본 연구의 목표는 가중치 변화를 정확하게 이해하고 시각화하는 것이기 때문에, 우선 각 epoch 별 가중치 파일을 기록하였다. 이렇게 저장된 가중치 파일을 바탕으로 가중치 변화를 분석하고 시각화하였다.

먼저 가중치 변화를 보다 면밀하게 관찰하기 위해 가중치 간의 상대적 차이를 구하여 표현하였다. 이를 위해 각 epoch 별 가중치와 이전 epoch의 가중치 차이를 절댓값으로 변환하여 저장하였다. 이로 인해 한 번의 epoch 당 가중치가 줄어들든 늘어나든 상관없이 가장 많이 변화한 레이어를 쉽게 알 수 있게 되었다. 이와 같은 계산을 계속 절댓값의 합으로 누적시켜서 누적 가중치 변화의 패턴을 관찰하여 학습이 계속 진행됨에 따라 가장 많은 변화를 일으키는 가중치는 무엇인지, 평균적으로 가장 많은 변화가 일어나는 레이어는 어디 인지를 체크할 수 있게 하였다.

또한 가중치들의 누적합을 모두 구한 후, 가장 큰 변화를 보여주는 가중치 값으로 모든 가중치 값을 나누어 0에서 1사의 값으로 모두 정규화 하였다. 이렇게 정규화 된 값은 크기에 따라 색으로 표현하여 시간의 흐름에 따라 밝아지는 레이어 색상을 통해 각 레이어들을 비교할 수 있도록 설계하였다.

이를 통해 가중치 변화에 영향을 미치는 여러 요소를 분석하는 것이 가능하였다. 예를 들어, 학습률과 최적화 알고리즘, 가중치 초기화 방법 등이 가중치 변화 패턴에 어떠한 영향을 미치는지 분석해 볼 수 있다. 또한 가중치가 적게 변하는 레이어의 크기를 바꿔 본다거나 아예 레이어 자체를 제거하거나 추가해 보며 이를 통해 모델 학습 전략을 조절하여 성능 향상을 도모할 수 있을 것으로 예상된다.

3.3 시각화 환경 설계

시간에 흐름에 따라 모든 레이어가 어떻게 변하는지 한눈에 살펴보기에는 3차원 환경이 적합하다고 판단했다. 2차원 상에서의 시각화에서는 레이어가 복잡하고 많아질수록 모든 레이어를 한 번에 보고 비교하는 것이 어려워진다. 따라서 본 연구에서는 3차원 공간 상에 레이어 구조를 대략적으로 표시하여 가중치의 상황을 시간의 흐름에 따라서 변하는 모습을 한눈에 볼 수 있도록 구성하였다.

Convolution Layer의 경우, 하나의 필터를 하나의 기준으로 잡고 출력 채널을 가로로 차원을 세로로 분리하여 하나의 층으로 구성하여 시각화하였다. 반면, Fully Connected Layer의 경우 기본적으로 가중치의 개수가 많아서 하나의 가중치 크기를 극단적으로 축소하고 임의의 기준으로 가로와 세로의 길이를 맞추어서 공간상에 표현하였다. (Figure 2 참조)공간 상에 레이어들을 표시할 때는 일정 간격을 두고 생성시키며 카메라가 3차원 공간상을 자유롭게 움직여서 자신이 원하는 방향에서 원하는 가중치 중심적으로 볼 수 있게 구상하였다. (Figure 3 참조)

또한, 현재 레이어의 최댓값과 평균값을 수치로 표시함으로써 각 레이어의 가중치를 수치적으로 비교할 수 있도록 하였다. 3차원 상에서 자세히 볼 수 없는 상황을 대비하여, 별도의 2개의 윈도우를 구성하여 2차원 상에서도 시각화를 자세히 볼 수 있도록 설계하였다. 이렇게 구축된 시각화 환경은 레이어 간 가중치 비교를 효과적으로 수행할 수 있게 도와줄 것으로 기대된다.

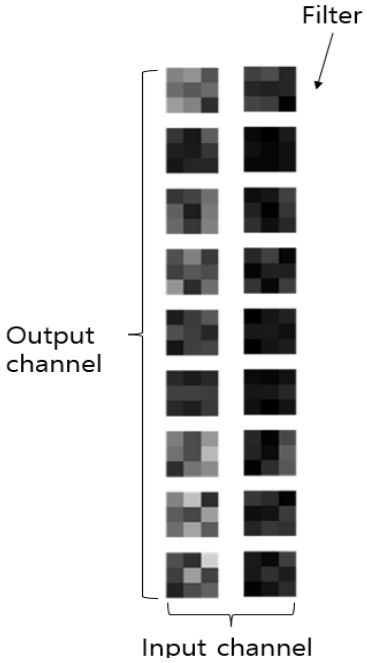

Figure 2: How to visualize the convolutional layer in this paper system.

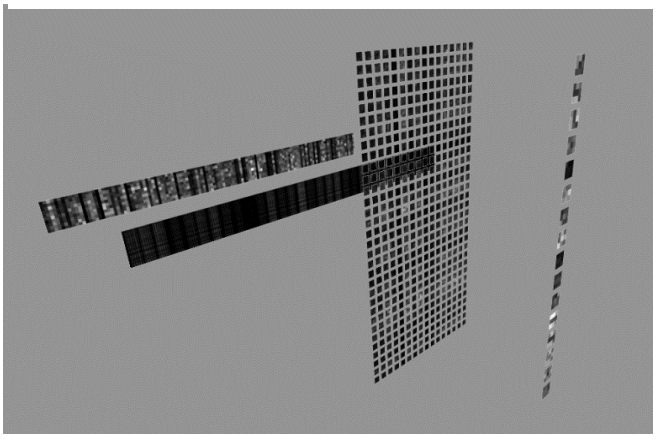

Figure 3: How to visualize the layers into 3D dimension in this paper system.

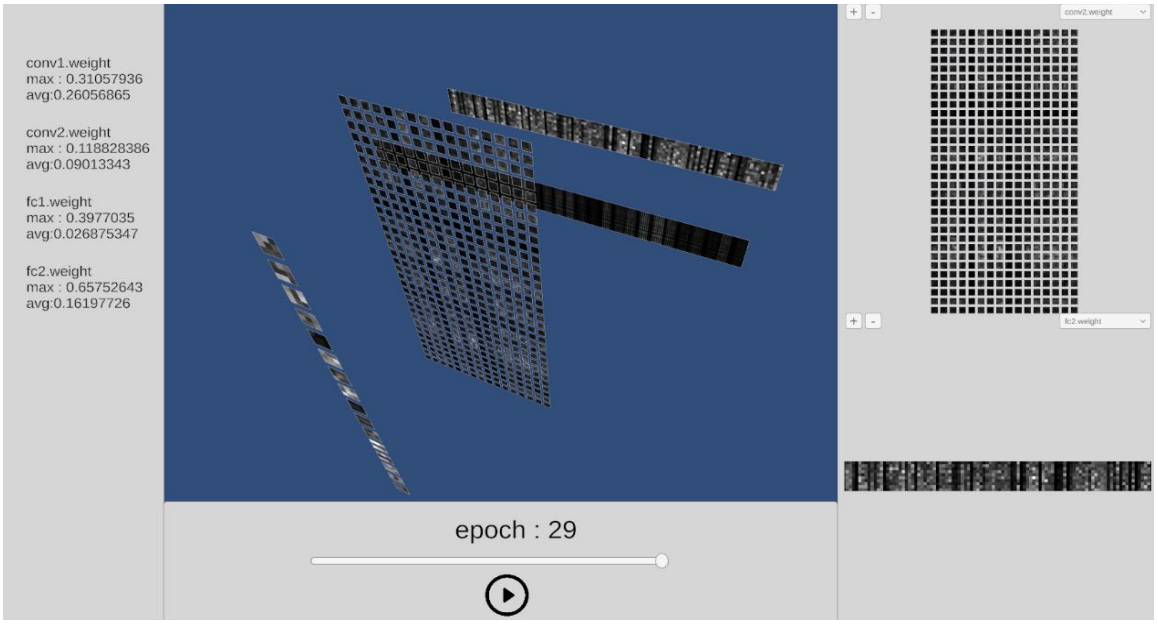

Figure 4: User interface of system in this paper

4. 실험 결과

4.1 시스템 구현

가중치를 시각화하는 프로그램을 개발하기 위해서 학습이 가능한 모델이 필요했다. 실험 진행을 위해 일반적으로 널리 사용되는 MNIST 데이터셋을 활용하였고, 파이썬 3.9 버전 기반의 환경에서 PyTorch를 사용하여 모델 학습을 진행했다. 또한, 학습 과정에서 얻어진 가중치를 시각화하기 위해 numpy 라이브러리를 이용하여 epoch 마다 가중치를 저장하고 레이어마다 바로 전 epoch 과의 차이를 구해서 차이의 절댓값을 누적하여 저장한 후, 누적된 값이 가장 큰 가중치 값으로 모든 가중치를 나눠서 정규화 시켰다. 그 후 시각화할 때 Matplotlib 라이브러리를 활용하여 0~1의 값을 0에 가까울수록 검은색 1에 가까울수록 흰색으로 표시하여 시간에 지남에 따라 색이 변하는 가중치를 표시하였다.

Unity 프로그램상에서는 프로그램을 시작 시 모든 가중치 값을 받아서 미리 Material로 만들어 두고 원하는 타이밍에 호출할 수 있도록 설정하고 첫 번째 가중치 값을 기준으로 레이어를 제작하여 일정 거리를 두고 카메라 앞에서부터 레이어 층을 생성하기 시작한다. 3차원 공간상에 레이어가 모두 배치되면 사용자의 시선이 되는 카메라는 키보드의 버튼을 통해 자유롭게 움직일 수 있어서 원하는 값을 중점적으로 볼 수 있도록 제작하였다. 좌측 창에는 현재 epoch에서의 각 레이어의 최댓값과 평균값을 나타내어 학습의 진행도에 따라 레이어의 상태를 정확한 수치로 볼 수 있고 다른 레이어와의 비교를 할 수 있도록 제작했다. 우측에는 2개의 창을 만들어 각각 원하는 레이어를 집중적으로 볼 수 있도록 만들었고 확대와 이동을 통해 원하는 부분을 좀 더 집중적으로 볼 수 있으며 언제든 다시 원하는 레이어로 변경이 가능하다. 아랫부분에는 epoch에 따른 시간의 흐름을 표시할 수 있게 만들어서 고정적으로 시간을 흐르게 할 수도 있고 자신이 원하는 시간의 값으로 바로 볼 수도 있다. 시간을 고정적으로 흐르게 한다면 시간의 흐름에 따라 변하는 가중치를 쉽게 판단하며 볼 수 있도록 제작했다. (Figure 4 참조)

4.2 실험 데이터 확인

본 프로그램이 잘 작동하는지 검증하기 위해 2개의 Convolution layer층과 2개의 Fully Connected Layer로 이루어진 환경에서 MNIST 데이터셋을 이용하여 실험을 진행했다.

Windows 환경에서 학습을 시켰으며 i7-12700(12 core, 20 thread, 4.9GHz) 프로세서와 GTX 3080 GPU, 32GB 이상의 메모리가 장착된 기기를 사용하였다. 학습 시간은 대략 10분 내외로 측정되었으며 시각화 프로그램을 작동 시 부팅 시간은 대략 3분 정도 소요되었다.

그 결과 프로그램 재생 시에 딜레이 없이 재생되었으며 학습에 따라 변화하는 가중치를 epoch 별로 관찰하여 가장 많이 변하는 값과 시기를 알 수 있었다. 실험상에서 가장 크게 변한 가중치는 두 번째 Fully Connected 레이어에 존재하는 가중치로 초깃값에 비해 0.6575의 변화율을 보였다. (Figure 8 참조) 평균적으로 가장 많이 변한 레이어는 첫 번째 Convolution 레이어 층으로 평균 0.2605의 변화율을 보였다. (Figure 5 참조) 가장 변화가

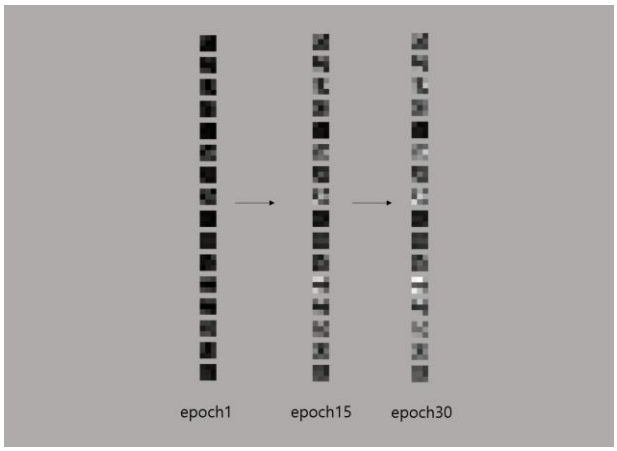

Figure 5: The change of convolution layer 1 through learning progress

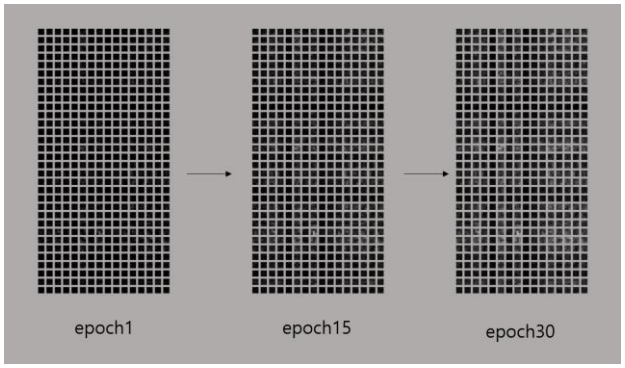

Figure 6: The change of convolution layer 2 through learning progress

Figure 7: The change of fully connected layer 1 through learning progress

Figure 8: The change of fully connected layer 2 through learning progress

없었던 레이어는 첫번째 Fully Connected 레이어 층으로 평균 변화율이 0.0268로 첫 번째 Convolution 층과 10배에 가까운 차이를 보였다. (Figure 7 참조)

두번째 Convolution layer층 같은 경우에는 평균변화율은 0.0900으로 첫번째 Fully Conneted보단 높지만 가장 많이 변한 가중치값이 첫번째 Fully Conneted는 0.3977 인데 비해 두번째 Convolution layer는 0.1188로 크게 변한 값이 없다는 걸 알 수 있다. (Figure 6 참조)

이를 토대로 가장 적게 변한 레이어 층의 구조를 바꾸거나 레이어 층을 늘리는 등 변경해야 할 하이퍼 파라미터에 대해 방향을 제시하는 자료로 충분히 사용할 수 있을 것이다.

5. 한계 및 제약사항

본 연구에서 개발된 프로그램은 네트워크가 학습하는 동안 가중치의 변화를 시각화하며, 이를 3차원 공간에서 확인할 수 있도록 설계되었다. 그러나 이 프로그램은 다음과 같은 제약사항과 한계점을 가지고 있다.

첫 번째로, 프로그램은 CNN 기반의 신경망에만 적용 가능하다. 이는 CNN 구조를 사용한 신경망에 대한 가중치 변화 표현에 초점을 맞춘 설계에 의한 결과이다. 다양한 신경망 구조에 대한 대응을 위해 프로그램을 확장하거나 수정할 필요가 있다.

두 번째로, 본 프로그램은 PyTorch 환경에서 개발되었기 때문에, PyTorch에서 작성된 인공지능 네트워크라면 모두 호환이 되도록 작성되었으나 다른 딥러닝 프레임워크에서는 작동하지 않는다. 이는 프로그램 구조 내에서 PyTorch 라이브러리의 함수와 객체를 사용하기 때문이다. 더 넓은 범위의 프레임워크 호환성을 위해 다른 딥러닝 프레임워크에도 대응할 수 있는 방식으로 프로그램을 재구성해야 한다. 이와 같은 제약사항 및 한계점을 극복함으로써, 다양한 신경망 구조와 딥러닝 프레임워크에서 가중치 변화를 시각화 할 수 있는 범용 도구의 개발이 기대된다. 이렇게 개선된 도구는 더 넓은 범위의 연구자와 개발자에게 도움을 제공할 것으로 예상된다.

6. 결론 및 향후 연구 제언

본 연구는 인공지능의 학습이 진행됨에 따라 변하는 가중치를 사용자로 하여금 확인할 수 있도록 하고 하이퍼 파라미터를 변경하는 데 있어 하나의 지표를 제시하기 위하여 누적 가중치를 시각화하는 도구를 제안하였다.

파이썬에서 PyTorch 를 활용하여 가중치를 epoch 별로 저장하였고, 바로 전 epoch 과의 차이를 저장 후 정규화 시킨 값을 시각화하여 학습 시 가중치가 얼마나 많이 변화하는지에 대한 자료를 제공하였다. 이 자료들은 Unity 에서 3차원 공간상에서 보이게 되어, 한눈에 학습이 진행됨에 따라 변하는 가중치의 모습을 확인할 수 있게 되었다.

다만, 본 실험의 시스템은 현재는 PyTorch 환경에서만 작동하며 CNN 네트워크 기반으로 작성되어 있어, RNN 등 다른 네트워크를 기반으로 한 시각화에 어려움이 있다. 이를 해결하는 방안으로 이를 해결하기 위한 향후 연구 방향으로는 ONNX 와 같은 통합 라이브러리를 활용하여 환경에 구애를 받지 않고 모두 사용할 수 있도록 개선할 수 있을 것이다. 또한 Saliency 기법 등을 활용하여 시각화를 진행한다면 CNN 뿐만 아니라 다른 네트워크 상에서도 시간에 흐름에 따른 시각화가 가능할 것으로 예상된다[10].
그러나 위에서 언급한 한계점에도 불구하고, 본 연구에서 제안한 시스템은 여태껏 무작위로 변경해 왔던 레이어의 개수와 가중치의 개수를 바꾸는 방향성을 제시하는 데 의의가 있다고 할 수 있다. 본 시스템을 이용하여 하이퍼 파라미터 변경에 대한 방향성이 도출되고, 이러한 연구가 지속된다면 더 높은 성능의 모델 개발에 기여할 것으로 기대된다.

References

[1] Tjoa, Erico, and Cuntai Guan. "A survey on explainable artificial intelligence (xai): Toward medical xai." *IEEE transactions on neural networks and learning systems* 32.11 (2020): 4793-4813.

[2] Arrieta, Alejandro Barredo, et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." *Information fusion* 58 (2020): 82-115.

[3] Van der Velden, Bas HM, et al. "Explainable artificial intelligence (XAI) in deep learning-based medical image analysis." *Medical Image Analysis* (2022): 102470.

[4] Zhou, Bolei, et al. "Learning deep features for discriminative localization." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[5] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016.

[6] Kumar, Siddharth Krishna. "On weight initialization in deep neural networks." *arXiv preprint arXiv:1704.08863* (2017).

[7] Bergstra, James, et al. "Algorithms for hyper-parameter optimization." *Advances in neural information processing systems* 24 (2011).

[8] Liashchynskyi, Petro, and Pavlo Liashchynskyi. "Grid search, random search, genetic algorithm: a big comparison for NAS." *arXiv preprint arXiv:1912.06059* (2019).

[9] Frazier, Peter I. "A tutorial on Bayesian optimization." *arXiv preprint arXiv:1807.02811* (2018).

[10] Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps." *arXiv preprint arXiv:1312.6034* (2013).

.

〈저자소개〉

양 태 린
- 숭실대학교 글로벌 미디어학과 졸업
- 숭실대학교 대학원 미디어학과 재학중(3학기)
- 그리다에너지 재직중
- https://orcid.org/0009-0000-9419-0690

박 진 호
- 1999년 2월 KAIST 수학과 졸업(학사)
- 2001년 2월 KAIST 응용수학과 졸업(석사)
- 2007년 8월 KAIST 전산학과 졸업(박사)
- 2013년 3월~현재 숭실대학교
  글로벌미디어학부 교수
- 관심분야: VR/AR, 딥러닝, 시각화
- https://orcid.org/0000-0002-4212-8382