AI의 재귀 학습의 최대 성능 및 붕괴 임계점 분석

2511 명진우, 2718 장우주
지도 교사 : 김하은

본 연구는 AI 모델이 자가 생성한 데이터를 재귀적으로 학습할 경우 발생하는 성능 저하와 붕괴 현상의 존재성과 구조를 실험적으로 규명하는 데 목적을 둔다. 이를 위해 시계열 예측에 강점을 가진 LSTM 모델을 기반으로, 1세대부터 최대 45세대까지 반복 학습을 수행하고 다양한 손실함수를 통해 세대별 성능을 정량적으로 평가하였다.

실험 결과, 초기 3세대까지는 높은 예측 성능을 유지하였으나, 4세대부터 손실함수의 급격한 상승이 관찰되었고, 이후 7세대 단위로 반복적인 과적합과 성능 붕괴가 발생하였다. 특히 20세대 이후에는 손실함수의 극대치가 점진적으로 상승하는 양상이 뚜렷하게 나타났으며, 이는 모델이 과도한 재귀 학습을 통해 정보 다양성을 상실하고 오차를 누적 학습하는 붕괴 구조에 도달한 것으로 해석된다. 또한, 인간 생성한 데이터와 AI가 생성 데이터의 비율을 조정하여 이분 탐색한 결과, 전체 데이터 중 약 87.5%를 인간 데이터로 구성했을 때 가장 낮은 오차율과 높은 예측 정확도를 보였다. 반면 인간 데이터 비율이 지나치게 높거나 낮을 경우, 성능 저하와 과적합 현상이 나타났다.

본 연구는 AI 모델의 재귀 학습이 성능 붕괴를 유발할 수 있다는 점을 실증적으로 입증했으며, 학습 세대 수와 인간-AI 데이터 비율 간의 최적 구간이 존재함을 밝혔다. 본 결과는 향후 범용 AI 모델을 개발하는 데 AI 붕괴의 방지 지표로써 활용될 수 있다. 다만 범용 모델에 사용되는 고급 알고리즘에 대해 분석을 진행하지 못한 점, 데이터 형식이 다원화된 멀티모달 등에 적용하지 못한 점 등은 후속 연구를 통해 검증이 필요하다.

1. 서론

최근 기계학습과 대형 언어 모델(Large Language Models, 이하 LLM)의 급속한 발전에 따라, 인공지능은 산업 전반에 걸쳐 폭넓게 활용되고 있다. 특히 ChatGPT, DeepSeek, Gemini와 같은 최신 LLM은 자연어를 입력받아 자연어로 응답하거나, 인간이 직관적으로 이해할 수 있는 다양한 형태의 결과를 생성한다.

최근에는 모델 컨텍스트 프로토콜(Model Context Protocol, MCP) 및 멀티모달(Multimodal) 기능의 고도화를 통해, 기존의 텍스트 입력·출력에 국한되지 않고 이미지, 영상, 음성 등 다양한 형태의 데이터를 입·출력할 수 있는 범용 인공지능의 형태로 진화하고 있다(김지원 외, 2015).

이러한 모델들은 대규모 인터넷 데이터를 기반으로 학습되며, 웹 페이지에서 자동으로 정보를 수집하는 크롤링 기술을 통해 데이터를 축적한다. 특히 LLM의 경우, 웹상에 존재하는 방대한 자연어 문서, 뉴스, 논문, 블로그, 코드 등을 크롤링하여 사전학습을 수행해 자연어 처리 및 생성 능력을 향상한다.

그림 1 시간에 따른 접속량 500개 사이트의 AI 크롤링 봇 트래픽 합. 2024년 1월부터 급격히 증가하는 모습을 볼 수 있다.

그러나 자동 수집을 기반으로 한 학습 구조는 부작용을 초래한다. 2024년 7월 기준 AI 크롤링 봇은 사이트별로 수천만에서 수억 건에 달하는 트래픽 요청을 유발하고 있으며, 이는 인간 사용자가 접근하는 상위 백만 개 웹사이트의 약 80%에 해당하는 수준이다(CloudFlare 보고서, 2024). 이 기능은 다양한 부작용을 초래할 수 있으며, 그중 대표적으로는 AI 붕괴가 꼽힌다.

AI 모델 붕괴란 AI가 재귀적으로 생성한 정보를 계속 학습하며 AI가 창의성을 잃거나, 특정 경향성으로 치우치게 되는 문제를 일컫는다(Ilia Shumaliov et al, 2024).

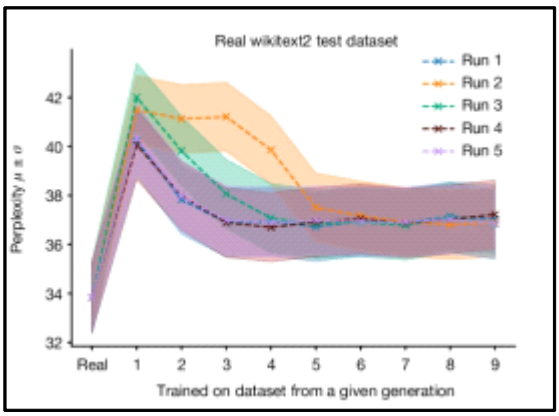

그림 1. AI 세대에 따른 단어 예측 성능(Perplexity) 비교. AI의 세대가 증가할수록 더 빠르게 낮은 값으로 수렴하는 경향을 띤다.

특히 LLM의 경우 LLM이 생성된 텍스트가 웹에 게시되고, 이 데이터를 크롤링해 다시 학습에 사용되는 순환적 구조가 구축되어 있다. 이는 AI가 정보를 선별하는 과정에 악영향을 미치며, 궁극적으로는 정보 다양성의 축소, 표현의 단조화, 그리고 특정 표현 패턴의 과잉 학습 등으로 이어질 수 있으며, 이는 곧 모델 붕괴 현상의 발생 가능성을 높인다(Elvis Dohmatob et al, 2024). 생성된 콘텐츠가 다시 학습 데이터로 활용되는 상황에서, 궁극적으로는 학습 데이터 전체가 생성 데이터로 오염되는 학습 생태계의 붕괴 문제로 확산할 수 있다.

AI 붕괴는 비단 LLM의 문제만 아닌, AI 모델 전체의 문제이다. Ilia Shumalov 외(2023)의 연구 결과에 따르면, Diffusion 모델이 스스로 생성한 이미지를 반복적으로 학습에 사용함으로써 발생하는 모델 붕괴 현상을 확인하였다. 연구진은 wikitext 기반의 CNN 모델을 구성하여, 다양한 자료형을 대상으로 초기 인간 데이터를 재귀적으로 학습해 AI 세대를 점차 증가시켰다. 세대가 거듭될수록 모델은 원래의 데이터 분포를 점차 손실하고, 결국에는 일관되며, 원형 자료형의 범주를 벗어난 자료만 제작하는 현상이 도출되었다.

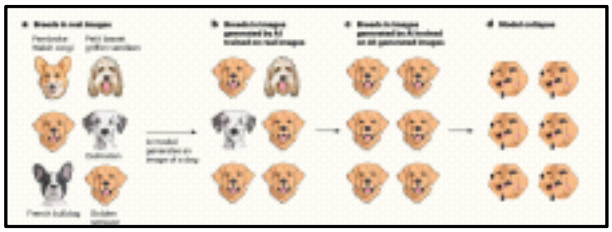

그림 3. 다양한 이미지를 대상으로 재귀적 학습을 진행하는 경우의 이미지 분포. 이미지가 하나의 종으로 수렴하더니, 결과적으론 원래의 데이터(강아지) 값을 소실한 모습을 볼 수 있다.

해당 문제는 CNN이나 Diffusion 모델에서만의 문제가 아닌, 다양한 AI의 종류를 가리지 않고 발생한다. 이는 교육용 기초 모델 수준에서조차 AI 붕괴의 위협에 놓여 있다는 의미이다. 궁극적으로 AI 붕괴로 인해 정상적인 자료수집 및 학습이 불가능해짐으로써 AI 성능 향상의 임계점을 맞이할 수 있다.

이러한 위험을 해결하기 위해, 본 연구는 AI 붕괴가 기초적인 모델에서 발생하는지 실증적으로 확인하고자 한다. 특히 시계열 학습에 최적화된 LSTM을 활용하여 AI 붕괴를 정량적으로 확인하고자 한다. 또한, AI에 의한 학습 데이터 증가 및 재귀학습에 따른 AI 모델 붕괴 현상을 고려한 최적의 인간-AI 데이터 비율을 도출함으로써, 장기적 모델 유지 관점에서의 최적의 데이터 활용 전략을 제시하고자 한다.

2. 이론적 배경

2.1 정규화

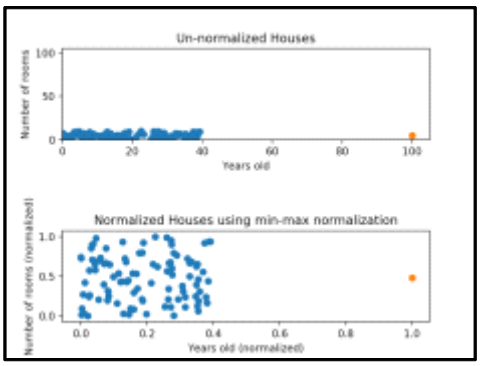

그림 4 정규화의 설명도. 최대-최소 정규화를 통해 모든 값을 [0, 1]로 제한한 모습이다.

정규화는 서로 다른 특성 및 범위의 값들이 같은 범위와 분포하도록 변환하는 과정으로, 머신러닝 및 딥러닝 모델의 학습 성능을 높이기 위한 데이터 전처리 기법의 하나다(네이버 국어사전, 2025). 모델을 구성할 시 학습용 변수 간 규모가 서로 차이가 나는 경우 경사 하강법 등의 학습 알고리즘이 정상적으로 작동할 수 없기에, 정규화를 통해 같은 분포로 통일시켜야 한다.

AI 분야에서 사용하는 정규화는 주로 표준 정규화, 최대-최소 정규화가 있으며, 각 모델의 목적 및 특성에 맞게 결정한다.

2.1.1. 최대-최소 정규화

최대-최소 정규화란 데이터의 값을 구간 [0,1]로 조정하여 데이터 간 규모를 통일하는 것에 목적을 갖는 정규화 방법이다.

$$x_{norm} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

데이터 간 규모가 같다면 경사 하강법, 가중치 선택 알고리즘 등을 수시 변화 없이 적용할 수 있어 모델의 학습 속도가 향상한다(Lucas et al, 2023).

그러나 최대-최소 정규화는 소수의 강한 이상값에 민감하다. 데이터 집합에 극단적인 값이 존재할 경우, 이러한 값들이 전체 데이터의 범위를 왜곡할 수 있다. 이에 따라, 정규화의 범위가 왜곡되어 AI의 성능 하락을 초래할 수 있다, 해당 문제를 방지하기 위해, Min-Max 정규화는 반드시 데이터를 전처리하는 과정을 거치고 사용되어야 한다.

2.1.2. 표준 정규화

표준 정규화란 데이터를 정규분포로 변환하는 방법이다. 식은 다음과 같다.

$$Z = \frac{X - m}{\sigma}$$

표준 정규화는 값을 모두 정규분포에 대응하여 변환하기에, 최대-최소 정규화에 비해 이상치에 대한 민감성이 낮으며 데이터의 범위에 미치는 영향이 상대적으로 적다.

그러나, 표준 정규화는 데이터의 범위가 정규분포 함수를 따르지 않거나, 데이터의 수가 적을 때 사용할 수 없으며, 특히 무작위적인 값에 대한 대응력이 떨어진다(씨익북스, 2025).

2.2 장단기 기억 네트워크(LSTM)

LSTM은 RNN의 한계를 보완하여 장기 의존성 문제를 해결할 수 있도록 작성된 딥러닝 모델 중 하나이다. LSTM은 RNN의 기본 구조에 몇 개의 추가적인 게이트를 제어하여 시계열 연산 등에서 강점을 보인다.

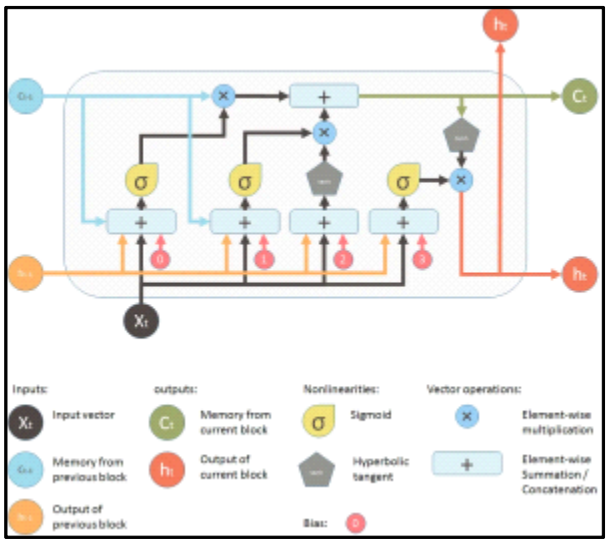

그림 5 장단기 기억 네트워크(LSTM)의 구조도. LSTM에서 각 계층의 상호작용을 보여준다.

LSTM의 핵심 구성 요소는 다층 구조의 레이어이며, 각 레이어는 입력 게이트(i_t), 망각 게이트(f_t), 출력 게이트(o_t), 셀 상태(C_t)로 구성된다.

2.2.1. 셀 상태

셀 상태는 LSTM의 내부의 메모리에 저장되는 값으로, 시계열 데이터의 장기 의존성을 구성한다. 셀 상태는 매 계층에 따라 변화하며, 이전 셀 상태는 현재 게이트에 입력값으로 작용한다.

2.2.2. 망각 게이트

망각 게이트는 LSTM 내에서 이전 셀 상태 중 어떤 정보를 유지하고 어떤 정보를 폐기할지를 결정하는 구조로, 셀 상태의 선택적 보존을 가능하게 한다. 보통 시그모이드 함수를 사용하여 각 요소의 보존 정도를 0과 1 사이의 값으로 계산한다. 이때, 시그모이드 함수의 값으로 0은 완전한 제거, 1은 완전한 보존을 의미한다.

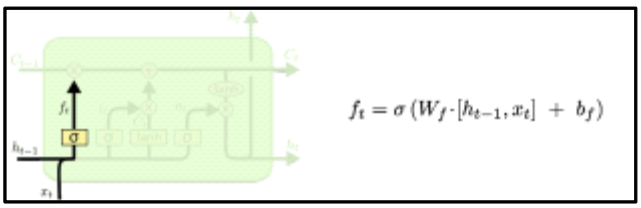

그림 6 망각 게이트의 개략도

해당 식에서 σ는 시그모이드 함수, W_f는 가중치 행렬, h_{t-1}는 이전 레이어의 은닉 상태, x_t는 셀 상태, b_f는 편향 값을 의미한다. 가중치 행렬과 편향 값은 AI의 모델에 따라 제작자가 임의로 결정한다.

이를 통해 망각 게이트는 특정한 편향 값과 과거 정보를 종합적으로 판단하여 셀 상태의 정보를 얼마나 보존할지를 조절한다.

2.2.3. 입력 게이트

입력 게이트는 새로운 입력값을 셀 상태로 변환하고, 기존 셀 상태와 비교할 수 있도록 데이터를 전처리하는 게이트이다. 새로운 데이터(x_t) 에 대해, 시그모이드 함수를 취해 각 벡터의 가중치를 판단한다. 그 후, 하이퍼볼릭 탄젠트 함수를 취해 각 벡터의 값을 기존 셀 상태의 값과 비교할 수 있도록 설정한다. 하이퍼볼릭 탄젠트 함수를 취한 값을 셀 후보라 한다. 셀 후보와 기존 셀 상태는 셀 상태에 추가될 정보의 내용을 결정한다.

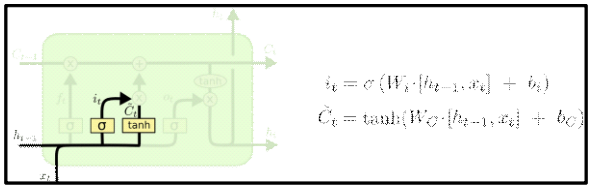

그림 7 입력 게이트의 개략도

이때, W_i는 입력 가중치 행렬, b_i는 입력 편향, $\tilde{C}_t$는 새로운 셀 상태, W_c는 후보군의 가중치 행렬, b_c는 후보군의 가중치 편향을 의미한다.

해당 식에 의해 셀 후보를 생성하고, 출력 게이트에서 셀 상태와 셀 후보를 비교한다.

2.2.4. 출력 게이트

출력 게이트는 LSTM에서 갱신된 셀 상태를 기반으로 현재 시점의 은닉 상태를 생성하여 다음 계층 또는 다음 시점으로 전달하는 역할을 한다. 기존 셀 상태의 모든 벡터값에 대해 하이퍼볼릭 탄젠트 함수를 취하여 범위를 [-1, 1]로 변경한다. 이후 시그모이드 함수를 계산하여 출력 활성화 비율(o_t)을 결정한다. 출력 활성화 비율은 새로운 셀 상태($\tilde{C}_t$)와 기존 셀 상태(C_t) 두 개의 대체 비율을 결정하여, 일부를 대체하여 새로운 셀 상태를 다음 계층으로 전달한다.

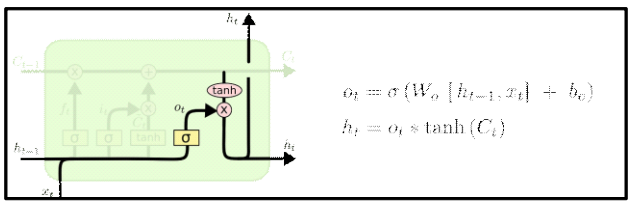

그림 8 출력 게이트의 개략도

이때, W_o는 출력 가중치 행렬, b_o는 출력 편향이다. 결과적으로, 셀 후보와 셀 상태가 일정 비율로 교체되어 새로운 셀 상태를 형성한다. 동기화된 셀 상태 및 새로운 은닉 상태는 다음 계층으로 넘어가 해당 게이트를 똑같이 통과한다.

2.2.5. LSTM의 특성

RNN과 대비되는 LSTM의 특징은 '시계열 데이터 적합도 증가', '기울기 소실 문제 해결' 이 있다.

2.2.5.1 시계열 데이터 예측률 증가

LSTM은 셀 상태에서 기존의 데이터를 참조, 재학습하므로 RNN에 비해 장단기 기억 보존에 유리하다. 특히 주식 모델, 강우 모델, 음성 인식 모델 등에서 RNN에 비해 뛰어난 성능을 보여왔다.

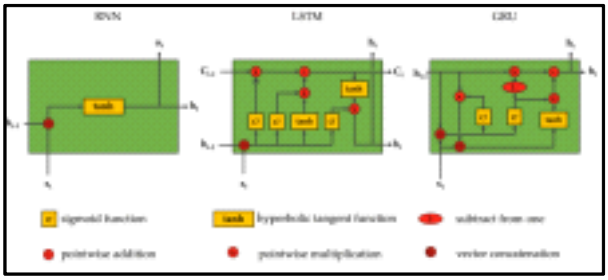

그림 9 RNN, LSTM, GRU의 구조도. GRU란 LSTM을 단순화한 형태의 AI 모델이다.

구조상 RNN은 단순한 구조로 이전 상태의 은닉층을 현재 입력과 함께 처리하는 순환 구조를 하고 있다. RNN의 구조는 계산 효율성 측면에서 장점이 있으나, 장기 의존성 문제로 인해 시퀀스가 길어질수록 초기 정보가 소실되는 한계가 있다.

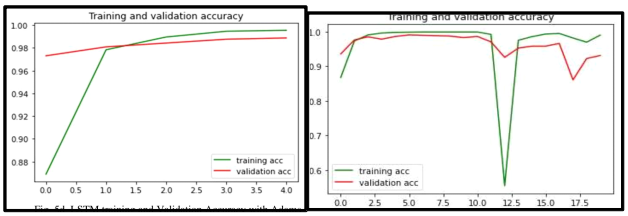

그림 10 LSTM 및 RNN에서 진행된 계층별 손실함수의 크기

LSTM은 RNN의 초기 정보 소실 문제를 해결할 수 있는 구조를 지녔다. 은닉, 입력, 출력 게이트는 기존의 정보를 참조함으로써 출력값을 동기화한다. 기존의 정보를 누적하여 참조하는 방식은 장기 의존성 문제를 해결할 수 있어, 시계열 예측 등 장기 데이터가 중요한 부분에 대해 효과적으로 대응할 수 있다.

Victor(2022) 외 연구에 따르면, 시간에 따른 스팸 메일 분류량에 대해 LSTM으로 예측을 수행했을 경우, RNN보다 최대 25% 높은 성능을 보여주었으며, 이는 다른 연구에서도 일관되게 관찰되는 현상이다. Pilla(2025)의 연구에서도 S&P 500 지수 예측에서 LSTM이 전통적인 RNN 모델보다 17.8% 높은 정확도를 보였으며, 특히 변동성이 큰 시장 상황에서 그 차이가 더욱 두드러지는 결과가 나타났다.

2.2.5.2. 기울기 소실 문제 해결

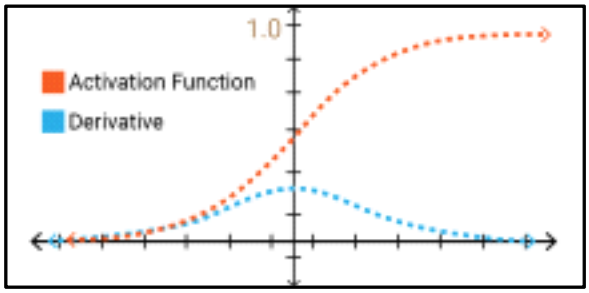

그림 11 기울기 소실 문제에 대한 모식도

LSTM은 RNN에 비해 기울기 소실 문제에 대한 정확성 하락 값이 적은 모델이다. 기울기 소실 문제란, 시퀀스가 길어질수록 역전파 과정에서 기울기가 점차 0으로 수렴하여 초기 레이어의 가중치가 효과적으로 동기화되지 않는 현상을 말한다.

RNN은 이전 데이터의 경향성을 확인할 수 없고, 바로 이전 데이터에 대한 정보만 활용할 수 있다. 따라서, RNN이 추측할 수 있는 가장 좋은 방법은 기존 데이터값과 비슷하게 추정하는 것이며, 이는 궁극적으로는 기울기 소실 문제를 유발하여 AI 성능을 제한시킨다.

LSTM은 이러한 RNN의 한계를 극복하기 위해 설계되었으며, 기존 셀 상태의 메모이제이션을 통해 셀 상태의 경향성을 참조하여 기울기 경향성을 정확히 파악할 수 있다. 특히, 망각, 입력, 출력 게이트는 연산 시 기존 셀 상태와 은닉 상태를 참조함으로써 정보의 추가, 제거, 출력 비율을 결정하여 ReLu 함수 및 활성화 함수의 기울기를 정밀하게 조절한다.

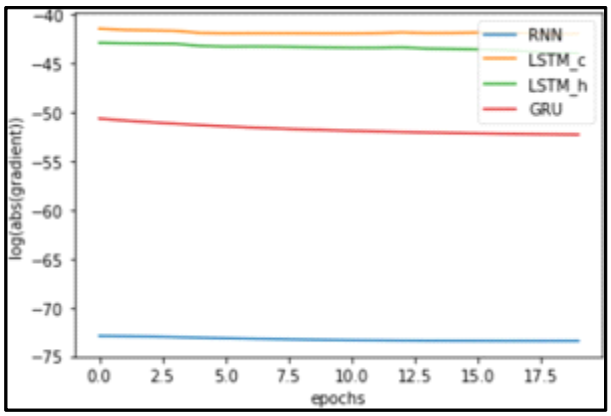

그림 12 Chen(2022)의 연구에서의 기울기 손실률. LSTM 모델이 전반적으로 높은 정확성을 가짐을 확인할 수 있다.

Hochreiter와 Schmidhuber(1997)의 논문에서는 LSTM이 RNN에 비해 100개 이상의 타임스텝에서도 기울기가 안정적으로 전파됨을 수학적으로 증명했

다. 또한, Chen(2022)의 연구에서는 기울기 흐름을 시각화하여 LSTM과 RNN을 비교했는데, 100 타임스텝 이후 RNN의 기울기 크기는 초깃값의 -75% 미만으로 감소하지만, LSTM은 초깃값의 60% 이상을 유지했다. 이는 LSTM의 셀 상태가 기울기 경로를 보존하는 데 얼마나 효과적인지 보여주는 증거이다.

2.3. 모델 붕괴

모델 붕괴란 AI가 재귀적으로 생성한 정보를 계속 학습하며 AI가 창의성을 잃거나, 특정 경향성으로 치우치게 되는 문제를 일컫는다(Ilia Shumailov 외(2023).

모델 붕괴는 통계 근사 오차, 과적합, 함수 표현력 오차라는 세 가지 결과를 갖는다.

통계 근사 오차란 AI가 값을 선별하고, 그 값을 재귀적으로 학습하는 과정에서 발생하는 오차이다. 세대가 거듭되어 재귀적으로 학습할수록 AI가 선별할 수 있는 변량의 수가 줄어들기에, 데이터의 부족으로 인해 AI 성능이 하락한다.

함수 표현력 오차란 이론적으로 정의된 함수가 컴퓨터 연산의 한계로 인해 오차가 누적되는 경우를 의미한다. 예를 들어, 정규분포 함수는 이론적으로는 함수가 개구간(-∞,∞)을 정의역으로 가져야 하나, 컴퓨팅 계산의 한계상 일정 범위로 치역을 제한하여 오차가 누적된다.

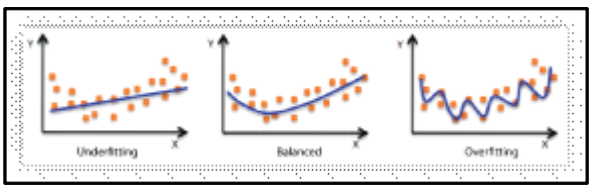

그림 13 과적합의 모식도. 우측 이미지가 데이터의 노이즈 경향성까지 학습한 모습을 보인다.

과적합이란 기계학습 모델이 훈련 데이터에 지나치게 최적화되어 새로운 데이터에 대한 일반화 능력이 저하되는 현상이다. 이는 훈련 데이터에 포함된 노이즈 또는 이상값 또한 학습하기 때문이다. 따라서, 훈련된 데이터에 대해 매우 높은 정확도 및 손실함수 수렴 속도를 보이나, 실사용 시 AI의 성능은 감소하는 경향을 보인다.

2.3.1. 모델 붕괴의 수학적 증명

이미 모델 붕괴에 관한 수학적 증명은 Lecheng Wang(2024) 등에서 활발하게 이루어졌으나, 정규분포 모델 및 기초적인 CNN 모델로 이를 국한하였다는 한계가 있다. 이러한 점을 보완하기 위해 본 연구자는 가우시안 분포로 해당 증명을 확장하여, 이론적 타당성을 보완하고자 한다.

항상 참인 데이터 X_0에 대해, $X_0 \sim N(\mu_0, \sigma^2)$이다. 해당 데이터를 이용하여 새로운 모델 M_1을 만들었다고 하자. 모델 M_1은 출력 데이터 X_1에 대해 평균과 분산을 추정한다.

$$\widehat{\mu_1} = \frac{1}{N}\sum_{i=1}^{N} x_i, \widehat{\sigma^2} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \widehat{\mu_1})^2$$

즉, 모델 M_1은 새로운 데이터 분포 X_1으로 새로운 데이터 분포 $X_1 \sim N(\mu_1, \sigma_1^2)$을 구성한다. 이를 반복하여 X_N을 구상하고 X_N의 분산 S_N은

이때 X_1의 분산 S_N은 분산 정리에 의해 다음과 같은 식을 만족한다.

$$S_N^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$$ 은 곧 분산의 정의이고,

$$\begin{aligned}\sum_{i=1}^{N}(x_i-\bar{x})^2 &= \sum_{i=1}^{N}[(x_i-\mu)-(\bar{x}-\mu)]^2 \\ &= \sum_{i=1}^{N}(x_i-\mu)^2 - 2(\bar{x}-\mu)\sum_{i=1}^{N}(x_i-\mu) + N(\bar{x}-\mu)^2 \\ &= \sum_{i=1}^{N}(x_i-\mu)^2 - 2N(\bar{x}-\mu)^2 + N(\bar{x}-\mu)^2 \\ &= \sum_{i=1}^{N}(x_i-\mu)^2 - N(\bar{x}-\mu)^2\end{aligned}$$

식 전개를 통해 다음과 같은 결과를 도출할 수 있다. 이때, μ는 모평균, $\bar{x}$는 표본평균을 의미한다.

한편,

$$\mathbb{E}\left[\sum_{i=1}^{N}(x_i-\mu)^2\right] = N\sigma^2$$

$$\mathbb{E}\left[N(\bar{x}-\mu)^2\right] = N\cdot \mathrm{Var}(\bar{x}) = N\cdot\frac{\sigma^2}{N} = \sigma^2$$

이 성립하므로(단, $Var()$는 분산 함수), 위에서 구한 식에 의해

$$\Rightarrow \mathbb{E}\left[\sum_{i=1}^{N}(x_i-\bar{x})^2\right] = N\sigma^2 - \sigma^2 = (N-1)\sigma^2$$

이 성립함을 알 수 있다. 따라서,

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2\right] = \frac{1}{N}(N-1)\sigma^2 = \left(1-\frac{1}{N}\right)\sigma^2$$

이다.

이는 가우시안 모델에 대해 데이터 학습 세대가 진행될 때마다 분산이 $(1-\frac{1}{N})$배 감소하는 것을 확인할 수 있으며, 이에 따른 AI의 데이터 선택 범위가 감소한다.

2.4. 손실함수

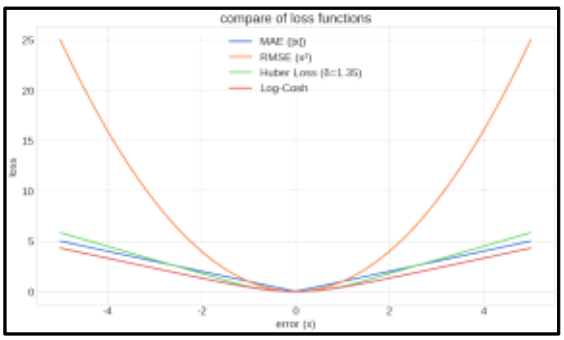

그림 19 오차값에 따른 손실함수의 크기

손실함수는 인공지능 모델이 선정한 예측값과 실제 관측값 간의 오차를 정량화하는 핵심적 평가 지표로서, 모델의 성능을 객관적으로 측정하고 비교하는 데 필수적인 요소이다. 모델 학습 과정에서는 대표적으로 RMSE, R^2, MAE, Huber Loss, Log-Cosh Loss의 5가지 손실함수가 사용된다.

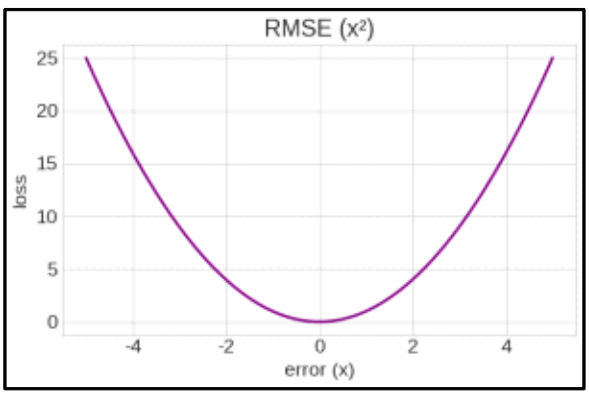

그림 20 오차값에 따른 손실함수의 크기

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$$

RMSE는 예측값과 실제값 간 차이의 제곱에 대한 평균의 제곱근으로, 오차의 크기를 원래 데이터와 같은 단위로 표현하는 장점이 있다. 그러나 제곱항으로 인해 수 개의 이상값에 대해 민감하게 반응하는 특성을 보기에, 불규칙적 데이터에서는 왜곡된 평가를 초래할 수 있다.

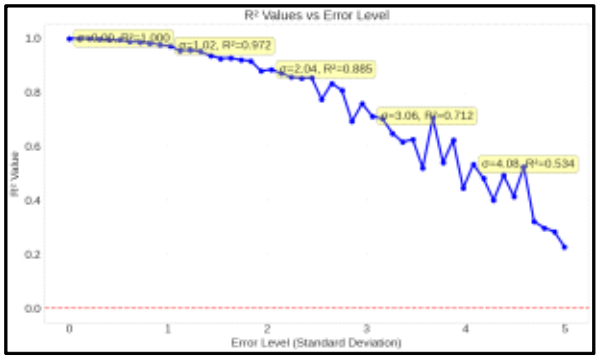

그림 21 정규분포 값에 따른 R^2값의 분포

$$R^2 = 1 - \frac{SSres}{SStot} (\text{단}, SS\text{는 제곱 합})$$

R^2은 모델이 설명하는 분산의 비율을 나타내는 지표로, 0에서 1 사이의 값을 가지며 1에 가까울수록 우수한 모델로 평가된다. 이 지표는 모델 및 다양한 실험의 정확도를 파악하는 데 대푯값으로 자주 사용된다.

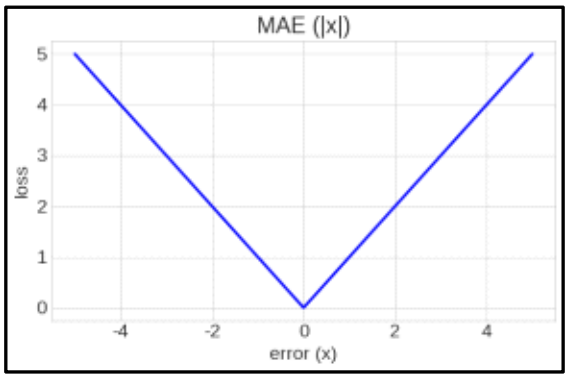

그림 22 오차값에 따른 손실함수의 크기

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i|^2$$

MAE는 예측값과 실제값 간 절대 오차의 평균으로, 오차의 방향성을 고려하지 않고 크기만을 평가한다. 절댓값 연산으로 인해 이상한지에 대한 민감도가 RMSE보다 낮아 불규칙적 데이터에서 RMSE에 비해 정량적인 성능 평가가 가능하다. 그러나 데이터가 불연속일 경우, 미분 불가능한 점이 필연적으로 발생한다. 이는 경사 하강법 기반 최적

화 과정에서 코드에 에러가 발생해 해당 알고리즘을 적용하지 못한다. 따라서 전처리 과정 등을 통해 불연속인 값을 조정해야 한다.

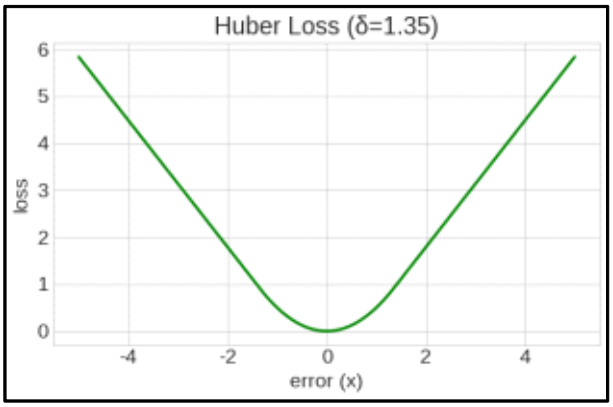

그림 23 오차값에 따른 손실함수의 크기

$$L_\delta(y,\hat{y}) = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & \text{if } |y-\hat{y}| \le \delta \\ \delta \cdot (|y-\hat{y}| - \frac{1}{2}\delta) & otherwise \end{cases}$$

Huber Loss는 MAE와 RMSE를 결합한 손실함수로, 임곗값(δ) 이하의 오차에 대해서는 MSE와 유사하게, 그 이상의 오차에 대해서는 MAE와 유사하게 작동한다. 일반적인 회귀분석 시에는 $\delta = 1.35$를 이용하며, 이는 RMSE와 MAE 사이의 중간 특징을 가진다. 이러한 특성으로 인해 이상치에 대한 견고성을 유지하면서도 작은 오차에 대해서는 미분 가능성을 보장하여 범용적인 오차 분석에 사용된다.

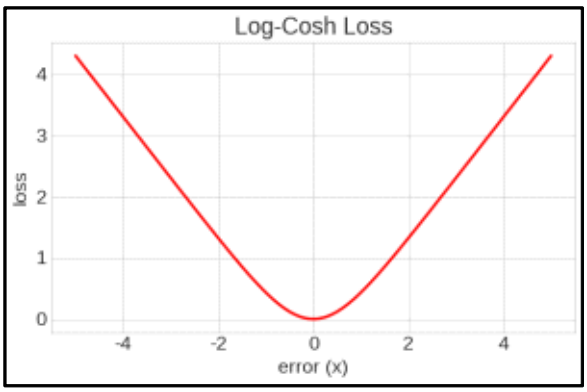

그림 24 오차값에 따른 손실함수의 크기

$$Log-Cosh(y,\hat{y}) = \sum_{i=1}^{n} \log(\cosh(\hat{y}i - y_i))$$

Log-Cosh Loss는 예측값과 실제값의 차이에 대한 쌍곡선 코사인에 로그를 취한 값으로, 작은 오차에 대해서는 RMSE와 유사하게 작동하나 큰 오차에 대해서는 강하게 증가하는 특성을 보인다. 이는 조금의 이상값에 대한 민감도를 낮추면서도 큰 이상값에 대해서는 전체 범위에서 미분할 수 있어 최적화 과정이 안정적으로 진행될 수 있게 한다. 특히 시계열 예측이나 회귀 문제에서 불규칙적 데이터를 다룰 때 유용한 손실함수로 평가된다.

3. 선행 연구 조사

반복 학습에 의한 AI 모델의 성능 저하 문제는 Ilia Shumailov(2024)의 'AI models collapse when trained on recursively generated data' 에서 핵심적으로 다루어진 바 있다. 이 연구는 대규모 언어 모델이 자체 생성 데이터를 반복적으로 학습할 경우, 정보 왜곡 및 표현력 붕괴로 이어지는 구조적 퇴화를 실증적으로 입증하였다. 특히 반복 세대를 거듭할수록 모델의 출력이 점차 현실과 동떨어지며, 언어적 다양성과 정확도가 감소하는 현상이 나타났다. 해당 연구는 재귀 학습이 모델의 일반화 능력에 부정적인 영향을 미칠 수 있다는 점을 강조하며, 반복 사용되는 데이터의 품질 저하가 핵심 원인으로 지목된다.

다만 이 논문은 실험 대상을 CNN 기반 Diffusion 기법에 한정하고 있으며, 시계열 데이터나 수치 예측과 같은 정량 분석 기반으로는 적용 범위가 제한적이다. 또한, 반복 학습에 따른 붕괴 현상의 기저 원리에 대한 수리적 분석보다는 현상 중심의 관찰에 중점을 두고 있어, 다양한 데이터 구조에 대한 일반화에는 한계가 존재한다.

이승연(2023) 저의 '인공지능 재귀 학습에 따른 모델 붕괴 현상 개선 방안' 에서는 반복적으로 생성된 데이터가 다양한 AI 모델의 분류 성능과 정보 처리 효율성에 미치는 영향을 여러모로 분석하였다. 특히 이미지, 자연어, 코드 등 여러 자료 형태에서 공통으로 관찰되는 성능 저하 양상을 통해 반복 학습의 범용적 위험성을 강조하였다.

본 연구는 반복 학습 시, 데이터의 정보 밀도 감소와 함께 모델이 고립된 표현 공간에 수렴하는 경향을 보이며, 이는 성능 붕괴를 일으키는 핵심 요인으로 해석했다. 또한, 기존 고품질 데이터의 재사용 없이 생성 데이터만으로 훈련을 지속할 경우, 모델이 초기에 습득한 일반화 능력을 점차 상실하게 된다는 점도 실험적으로 제시하였다.

그러나 이 논문 역시 주로 이산적 분류 문제에 초점을 맞추고 있어, 연속적인 시계열 예측이나 경제 데이터의 누적 왜곡에 대한 구체적인 분석은 부족하다. 또한, 반복 학습의 붕괴가 실제 예측 정확도에 미치는 영향에 대해선 구체적인 지표를 제시하지 않아, 정확한 값 비교가 불가능하다는 점에서 한계가 존재한다.

4. 연구의 목적

본 연구의 목적은 다음과 같은 세 가지 목적으로 기술된다.

3.1. LSTM 기반 AI 붕괴의 실험적 존재성 확인

LSTM을 여러 세대 만들고, 재귀적으로 학습하는 AI 모델을 여러 세대 제작하여, 전 세대의 데이터를 학습시키는 과정을 반복한다. 이를 통해 AI 모델이 재귀적으로 학습할 경우, 붕괴하는 현상을 실험적으로 확인한다.

3.2. AI 붕괴의 임계 세대 확인

AI 붕괴가 발생하는 임계 세대를 확인한다. 세대 변화에 따른 AI 성능 지표를 MSE, R^2값 등의 다양한 지표를 통해 비교하여, 그래프를 종합적으로 판단하여 AI 붕괴의 임계 세대를 확인한다.

3.3. AI 학습 시 효율적인 인간-AI 데이터 비율 확인

재귀적 학습에 의한 데이터 증폭 현상과 재귀적 학습에 따른 AI 모델 붕괴 현상을 종합적으로 고려하여, 가장 효율적인 인간-AI 데이터 비율을 확인한다. 구해낸 AI-인간 데이터 비율과 AI의 데이터를 주기적으로 학습한 경우의 성능을 비교하여 효율성을 입증한다.

5. 연구 과정

본 연구는 다음과 같이 가설 설정, 데이터 선정 및 전처리, LSTM 구현 및 1세대 생성, LSTM N세대 생성 및 재귀 학습 실행, 초기 데이터의 인간-AI 비율 변경으로 구성된다.

5.1. 가설 설정

본 연구는 다음과 같은 가설로 구성된다.

첫 번째, AI가 자가 생성 데이터로 학습을 진행하면서 일정 수준까지 정확도가 올라가다 그 뒤로 급격하게 감소할 것이다. Nautre에서 진행한 연구 등에 따르면, 자가 생성 데이터를 다시 학습에 사용할 일정 수준의 다양성 확보를 통해 AI의 성능이 올라간다고 제시하였다. 이에 따라, 일정 세대까지는 AI의 정확도가 상승할 것으로 예측하였다. 그러나 일정 수준이 지나면 AI가 자가 생성 데이터에 과도하게 학습되어 과적합이 발생할 것이다.

둘째, 인간 데이터와 AI 생성 데이터의 최적 비율은 약 20%일 것으로 예상된다. 인간 데이터 비율이 이보다 높아질 경우, 학습 효율성 증가가 둔화할 것이며, 반대로 AI 생성 데이터 비율이 지나치게 높아질 경우, AI 모델 붕괴 현상이 발생할 가능성이 있다. 따라서 모델의 안정적 성능과 학습 효율성을 모두 고려할 때, 약 20%의 인간 데이터 비율이 최적점일 것으로 가설을 설정한다.

5.2. 변인 통제

본 연구자는 1번 실험에 대해, 조작 변인으로서 AI의 세대를 선택하였으며, 종속 변인으로서는 각 매개변수의 손실함수, 통제 변인으로서는 그 외의 모든 실험 환경을 선택하였다. 또한, 대조군으로서는 1세대의 데이터를 활용할 것이다.

본 연구자는 2번 실험에 대해, 조작 변인으로서 인간-AI 데이터의 비율을 선택하였으며, 종속 변인으로서는 각 매개변수의 손실함수, 통제 변인으로서는 그 외의 모든 실험 환경을 선택하였다. 또한, 대조군으로서는 인간 100%의 데이터를 사용할 것이다.

5.3. 데이터 선정 및 전처리

본 연구에서는 UCI의 공유 자전거 데이터셋을 사용하였다. 이 데이터는 계절, 공휴일 여부, 온도, 습도 등의 기상 및 환경 정보와 시간대별 공유 자전거 이용 수를 포함하고 있다. 해당 데이터를 선택한 이유는 다음과 같다.

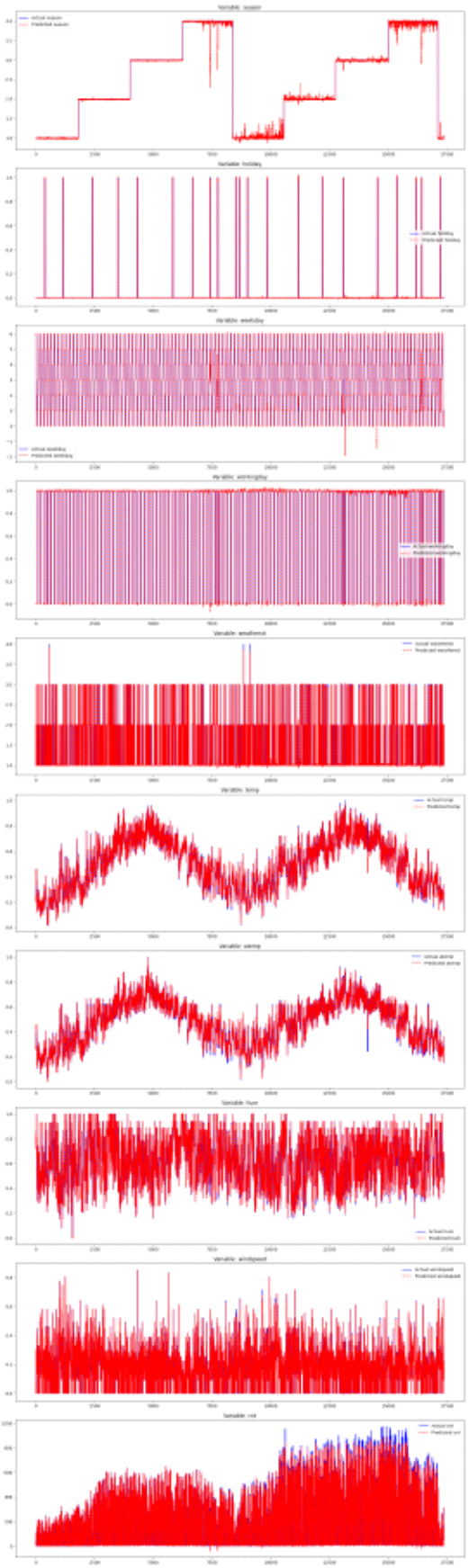

그림 25 UCI Bike Sharing Data set에 있는 데이터의 종류. 편의상 실험 결과 그래프를 사용했다.

첫째, 시간 단위로 약 2년간 기록된 17,000개 이상의 대규모 데이터를 포함하고 있어, 충분한 학습량을 제공하며 통계적 신뢰성이 높다.

둘째, 데이터의 형식이 다양하다. 예컨대, working day와 같은 주기성을 갖는 범주형 변수, yr 같은 시간 경과에 따라 단조 증가하는 변수, cnt, temp와 같은 연속형 변수가 혼재되어 있어, 붕괴 시점이나 학습 속도, 포화 여부 등 데이터 특성에 따른 모델의 반응을 관찰하고 분석하는 데 유리하다.

셋째, 해당 데이터셋은 이미 다양한 연구에 활용되어 그 일반성과 활용도가 검증되어 있다. 예를 들어, 김인희(2018)는 이 데이터를 이용해 LSTM과 랜덤 포레스트 모델의 성능을 비교하였으며, Albuquerque 등(2024)은 도시 교통 유동량 개선을 위한 머신러닝 모델 학습에 활용하였다. 또한, Pan 등(2018)은 심층 강화학습 기반의 도킹 없는 자전거 재배치 문제에 이 데이터를 적용하였다.

이러한 점에서 본 데이터셋은 다양한 머신러닝 기법을 실험하고 그 성능을 검증하기에 적합한 표준 벤치마크 데이터로 평가되어, 해당 데이터를 사용하였다.

5.4. 실험 과정

5.4.1. 학습 데이터 재정의

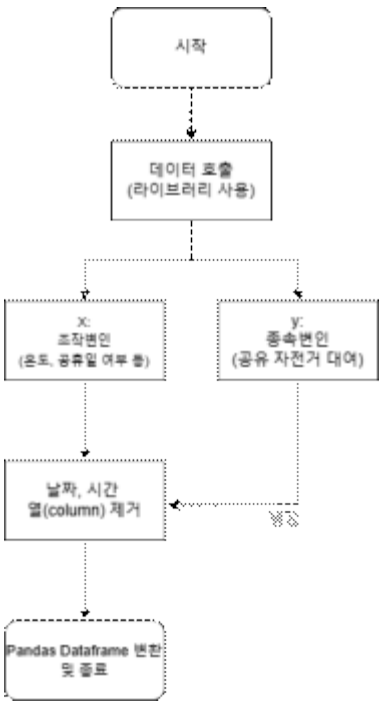

그림 26 학습 데이터의 재정의

해당 데이터 중 단순한 인덱싱을 포함하는 시간 데이터인 날짜(date), 시간(hr)의 데이터를 제거하는 전처리 과정을 거치고, 입력 변수에 대여 횟수(cnt)를 추가한다.

5.4.2. LSTM 1세대 생성

5.4.1.에서 생성한 데이터를 기준으로, 1세대의 LSTM을 생성한다. 1세대 LSTM은 기존의 항상 참값(실제 데이터)인 값을 학습하고, AI가 정상적으로 학습했는지 확인한다.

5.4.2.1. LSTM 구현

LSTM의 구현은 다음과 같다.

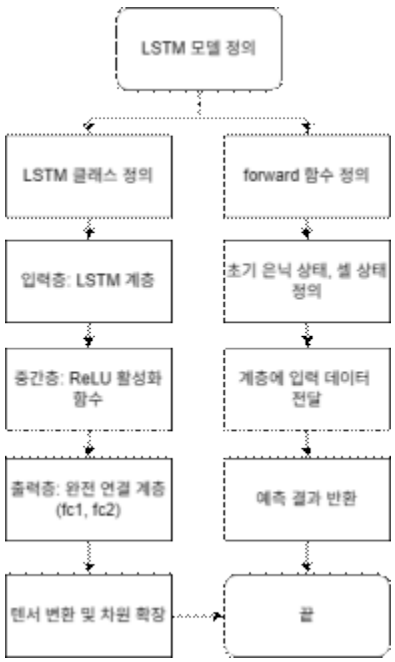

그림 27 개조된 LSTM의 클래스

모델을 호출하여 즉각적으로 사용할 수 있도록, LSTM을 클래스 형식으로 제작하였다. LSTM 클래스 내부에는 전파 연산을 수행하는 forward 함수를 함께 정의하여, 입력 데이터를 받아 예측 결과를 반환하는 흐름을 구성한다.

입력층에서는 시계열 데이터를 단위로 중간층 계층에 전달한다. 이때 중간층은 각 시점마다의 입력을 처리하면서, 내부적으로 은닉 상태를 주어진 횟수만큼 반복하여 데이터의 정확도를 높인다. 한 계층의 은닉 상태를 통과하면 다음 계층으로 이동하는데, 해당 계층에서는 은닉 상태에서 생성한 데이터를 손실함수가 적어지는 방향으로 재가공한다. 모델이 학습을 시작하기 전에는 이 두 상태를 0으로 초기화하며, 이를 통해 시계열 샘플 간의 독립성을 보장한다.

초기화된 은닉 상태 및 셀 상태와 함께 입력 데이터가 계층에 전달되면, LSTM은 이를 시점 단위로 순차적으로 처리하면서 시계열 내의 패턴을 학습한다. 해당 과정에서 AI 학습에 많이 사용되는 비선형 활성화 함수인 ReLU를 통과하여 중간층에서 변환된다.

활성화된 출력은 이후 완전 연결 계층을 거친다. 본 모델에서는 두 개의 완전 연결 계층(fc1, fc2)을 사용하여 시계열 특성 정보를 최종 예측값으로 변환한다. 이때 출력층의 구조는 예측하고자 하는 값의 종류에 따라 조정할 수 있으며, 회귀 문제의 경우 단일 값을 반환하도록 구성된다.

모델 전처리 과정에서는 입력 데이터가 LSTM 계층이 요구하는 형태인 3차원 텐서(batch size, sequence length, feature dimension)로 변환된다. forward 함수는 해당 과정을 통해 예측 결과를 반환하며, 이는 후속 손실함수 계산 및 역전파를 통해 모델의 파라미터 학습에 활용된다.

5.4.2.2. 라이브러리 호출 및 구글 드라이브 연결

LSTM을 사용하기 위한 라이브러리로서 Python의 torch, pandas, Tensor, math 라이브러리를 설치 및 호출한다.

5.4.2.3. CUDA 사용 설정

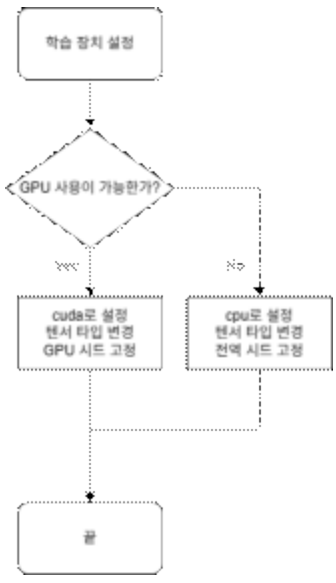

그림 28 CUDA 사용 설정

엔비디아의 GPU 병렬 처리 프로그램인 CUDA를 사용하기 위해, A100의 CUDA로 학습 장치를 재설정한다.

5.4.2.4. 데이터 정규화

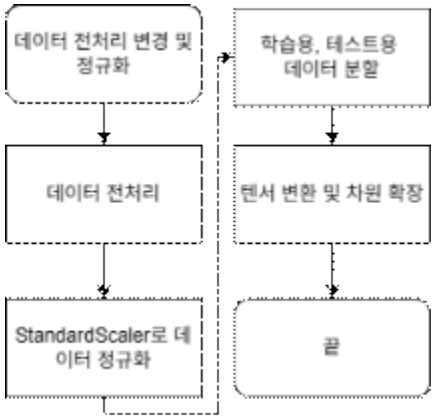

그림 29 데이터 전처리 수행

모델 학습 전, 텐서 변환 및 LSTM의 안정적인 학습을 위해 입력 데이터를 표준 스케일링 기법을 통해 정규화하였다. 표준화는 각 특성의 분포가 평균 0, 분산 1이 되도록 변환함으로써, 변수 간의 단위 차이나 스케일 차이로 인해 발생할 수 있는 학습 편향을 방지하고, 모든 입력값이 같은 가중치를 갖도록 유도한다.

모델이 학습하는 데이터와 테스트용 데이터를 분할 한다. 해당 모델은 입력과 출력의 형식이 일정하므로, 테스트용 데이터와 학습용 데이터 모두 X의 부분집합으로 처리한다. 학습 비율과 테스트의 비율은 1:1로 설정하였다.

이후 텐서 변환을 통해 LSTM이 입력받고, 역전파 알고리즘을 수행할 수 있도록 저장한다.

5.4.2.5. 하이퍼파라미터 조정 및 학습 시행

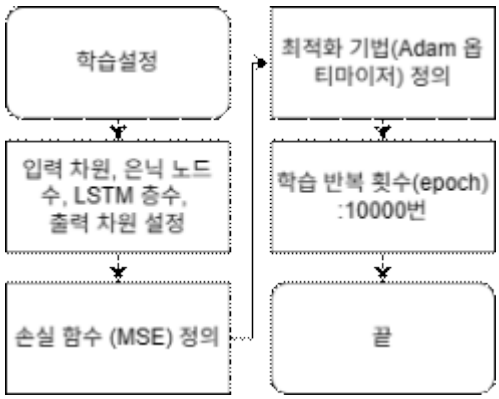

그림 30 전체적인 학습 과정

X를 기반으로 학습을 설정한다. 학습 데이터 중 은닉 계층은 Graves et al (2013)의 연구를 참조하여 epoch = 10000, layer = 5 hidden_size = 128로 설정하였다.

5.4.2.6. 모델 학습 과정

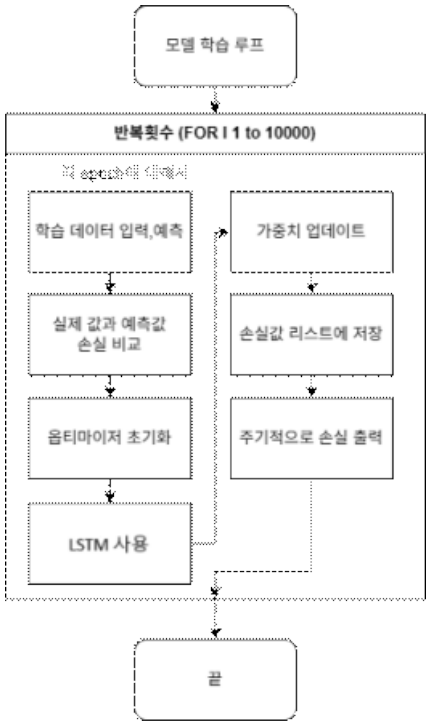

그림 31 각 학습 과정에서의 구조

각 학습 과정(epoch)은 다음과 같이 구성된다.

먼저 학습 데이터를 입력하여 예측값을 도출한 뒤, 실제값과 비교하여 손실을 계산한다. 이 손실 값은 모델의 성능을 정량적으로 평가하는 지표로 사용된다. 학습 과정 수가 증가할수록 손실함수가 감소하는 방향으로 진행된다. 본 연구에서는 첫째, 값의 단위와 특정 값이 가중되지 않는 평균 제곱 오차를 손실함수로 채택하였다. 이후 계산된 손실 값은 리스트에 저장되고, 일정 주기마다 출력되어 학습 진행 상황을 시각적으로 확인할 수 있도록 하였다.

손실 계산이 완료되면, Adam Optimizer를 초기화한 후, 손실을 최소화하기 위한 방향으로 모델의 가중치를 업데이트한다. 모든 업데이트가 완료된 후, 다음 epoch로 넘어가기 위해 LSTM 모델을 재사용하며, 위 과정이 설정된 반복 횟수만큼 반복된다.

5.5. N세대 LSTM 구성

N세대의 LSTM은 1세대 LSTM과 그 구조 및 기능이 대부분 같나, 다음과 같은 차이점이 있다.

첫째, 1세대 LSTM은 UCI에서 지정한 데이터셋을 호출하여 불러오지만, N세대는 $N-1$세대가 예측한 데이터셋을 사용한다.

둘째, 1세대 LSTM은 전체 데이터의 앞 50%가 학습 데이터, 뒤 50%로 구성된다. 그러나 N세대에서는 만약 n이 짝수라면 앞 50%가 학습 데이터, 그렇지 않을 경위 뒤 50%가 학습 데이터로써 사용된다.

5.6. 시각화 및 에러 출력 코드 작성

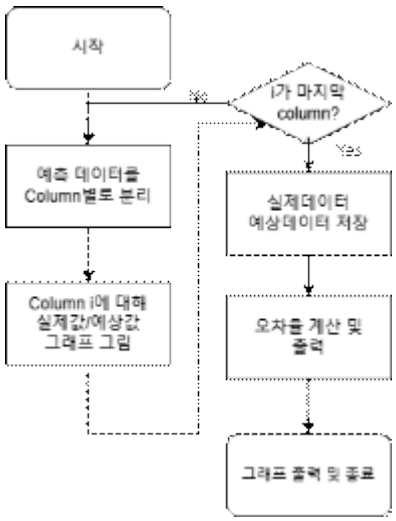

그림 32 시각화 코드의 순서도

시각화 코드는 다음과 같이 구성된다.

예측값과 실제값을 열에 따라 구분하고, 해당 값에 대해 실제값과 예상값을 비교한 그래프를 출력한다. 모든 변인에 대해 이를 시행하고, 에러를 출력한다. 그 후, 각 파일을 드라이브에 저장하여 다음 세대에서의 학습 데이터로서 작용하도록 한다.

5.7. 실험 과정

5.7.1. AI 붕괴 현상 확인 및 임계점 분석

본 실험은 다음과 같은 절차에 따라 수행되었다.

1. 1세대 LSTM 모델을 학습시켜 대조군을 확보한다.
2. 이후 같은 구조의 LSTM을 59회 반복 학습시켜, 총 45세대의 LSTM 학습 결과를 확보한다.
3. 세대별 모델의 예측값과 실제값 간의 오차를 계산하여 저장한다.
4. 저장된 오차 데이터를 시각화하여, 반복 세대에 따른 오차의 추이를 분석하고, 성능 붕괴가 시작되는 임계점을 확인한다.

5.7.2. 인간-AI 데이터 비율에 따른 적합성 분석

실험은 다음의 순서로 진행되었다.

1. 1세대 LSTM을 학습시켜 AI 100% 구성의 예측 데이터를 확보한다.
2. UCI Bike Sharing Dataset 원본 데이터를 참값(인간 데이터)으로 간주하여, 인간 100% 구성의 데이터를 수집한다.
3. 이후 학습 데이터 구성 비율을 이진 탐색 알고리즘을 활용하여 탐색한다.
4. 각 데이터 비율에 대해 같은 구조의 모델을 학습시키고, 예측값과 참값 간 오차를 10세대에 걸쳐 비교한다.
5. 전체 실험 결과 중 가장 오차가 낮은 데이터 비율을 최적 비율로 설정한다.

6. 실험 결과

실험 결과 전체는 하단 드라이브에서 확인할 수 있다.

6.1. 1번 실험 결론

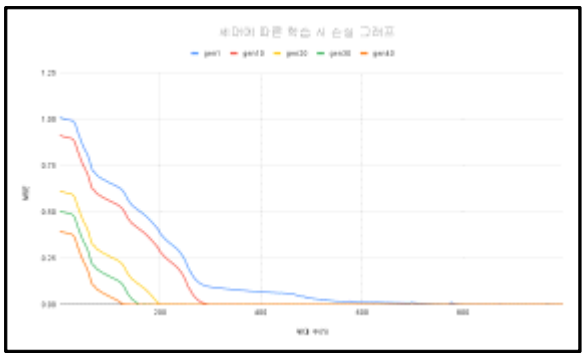

그림 33 세대에 따른 학습 시 손실 그래프

그림 34 재귀 학습 시 세대에 따른 손실함수의 정확도

6.2. 2번 실험 결론

그림 35 인간 비율이 93.5%일 때의 세대에 따른 손실함수

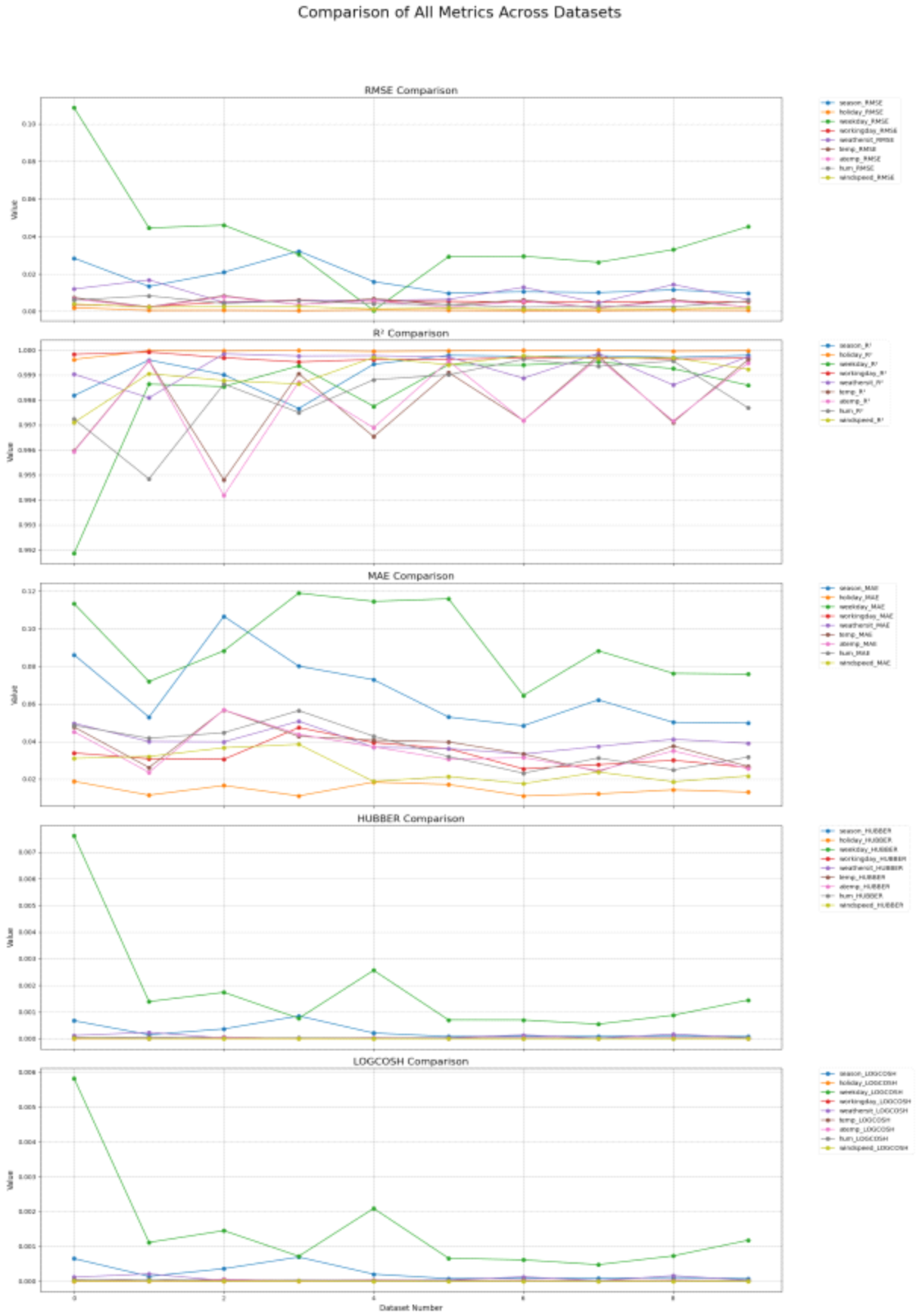

그림 36 인간 비율이 96%일 때의 세대에 따른 손실함수

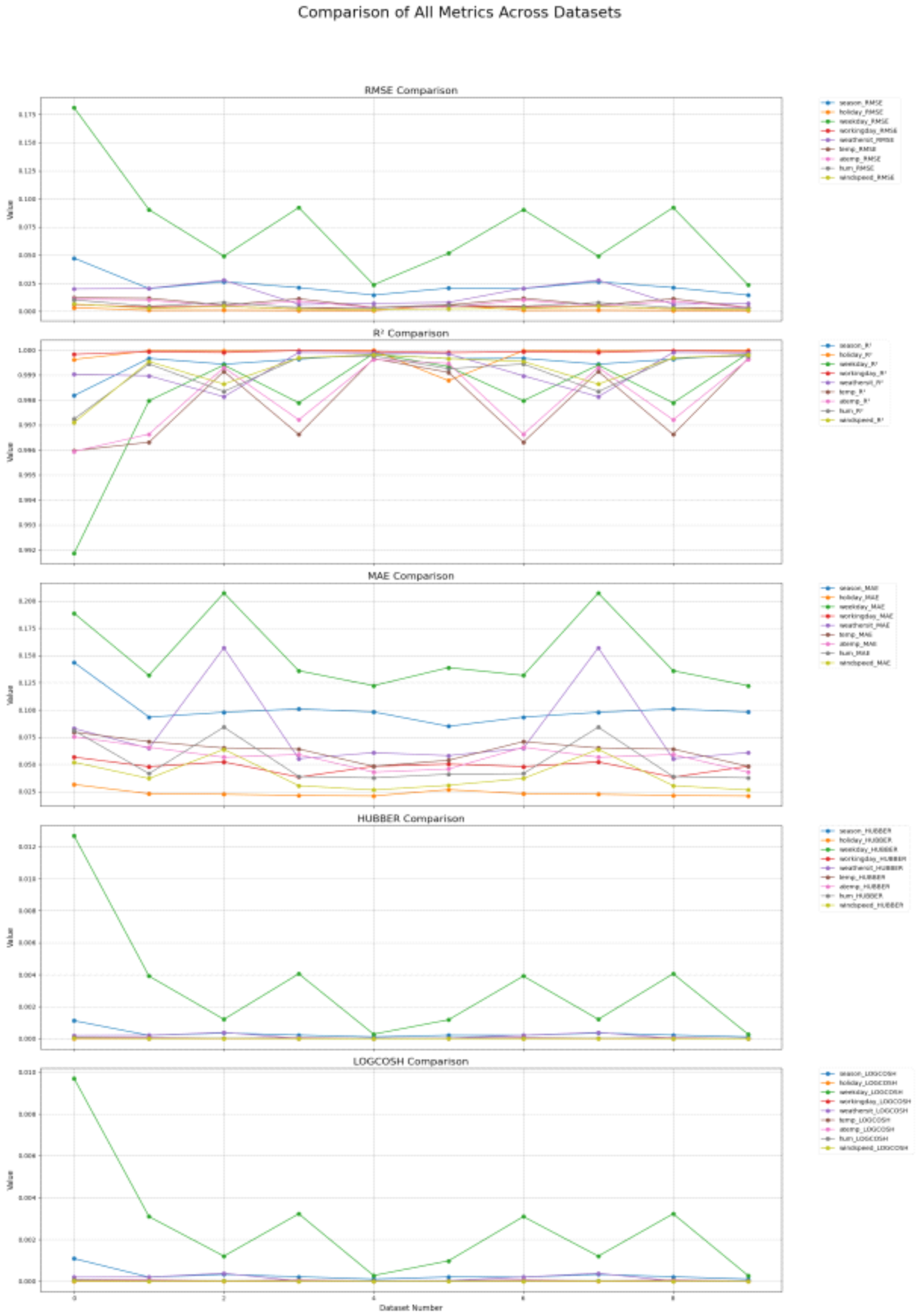

그림 37 인간 비율이 100%일 때의 세대에 따른 손실함수

7. 결론

7.1. 1번 연구

1번 실험에서는 재귀적 데이터를 활용한 조건에서 반복 학습에 따른 인공지능 모델의 성능 변화 및 잠재적 붕괴 가능성을 검증하고자 하였다. 총 45개의 주기적 구조를 가진 데이터셋을 대상으로 변수별 회귀 모델을 구성하고, 다양한 손실함수(MAE, RMSE, Huber, Log-Cosh)를 적용하여 정량적 평가를 시행하였다.

실험 결과, 초기 학습 구간에서는 전반적으로 높은 R^2 및 낮은 손실 함숫값이 약 3세대 동안 유지되었으나, 4세대와 6세대에서 손실함수의 크기가 급격히 증가하는 양상이 관찰되었다. 이후에도 약 13세대, 20세대 등 대략 7세대 주기로 손실함수가 증가하고 모델의 성능이 악화하는 징후가 지속해서 나타났다. 이러한 현상은 재귀적 학습 반복에 따른 과적합으로 해석될 수 있으며, 특히 weekday나 season과 같이 기본적으로 손실 함숫값이 큰 변수에서 더 두드러지게 발생하는 경향을 보였다.

20세대 이후부터는 주기적으로 반복되는 손실 함숫값의 극대치가 점진적으로 증가하는 양상이 확인되었다. 이전 최댓값(19세대)에 비해 손실함수의 크기가 평균 5%씩 증가하는 현상이 모든 변수에서 공통으로 관찰되었다. 또한, 세대가 반복됨에 따라 손실함수의 크기가 전반적으로 증가하는 경향이 나타났다. 특히 season, weekday, hum 변수는 R^2 지표 기준으로 후반 구간에서 0.92 이하로 하락하는 구간이 반복적으로 발생하였으며, MAE 및 RMSE 지표에서도 hum, atemp, temp 변수에서 손실함수의 크기가 점진적으로 증가하는 경향이 관찰되었다.

세대에 따른 학습 손실함수의 크기 변화를 분석한 결과, 세대가 진행됨에 따라 학습 손실함수의 크기가 급격히 감소하는 경향성이 확인되었다. 특히 30-40번째 구간에서 손실 값의 급격한 진동 및 예측 성능 저하가 집중적으로 발생하였다. 이는 단순한 학습 진동이 아닌, 누적된 학습 구조의 데이터 오차값까지 모두 학습하는 과적합 현상이 발생했음을 시사한다.

결론적으로, 본 실험을 통해 AI 붕괴 현상이 실증적으로 확인되었으며, 이는 약 20-30세대 학습 시점에서 뚜렷하게 나타났다. AI 붕괴의 주요 특징으로는 손실함수의 점진적 증가, 손실함수 극대치의 비율 증가, 주기적 과적합 현상이 관찰되었다. 이러한 결과는 'AI가 자가 생성 데이터로 학습을 진행하면서 일정 수준까지 정확도가 올라가다 그 뒤로 급격하게 감소할 것'이라는 초기 가설과는 일부 어긋난다. 실제로는 일정 수준까지 정확도가 지속해서 향상되지 않고, 주기적인 손실함수의 변동과 점진적인 성능 저하가 관찰되었다.

7.2. 2번 실험의 결론

본 연구의 두 번째 실험은 인간 데이터의 비율을 점진적으로 증가시키며, 예측 모델의 성능이 어떻게 변화하는지 다양한 손실함수로 분석하였다. 실험에서는 RMSE, MAE, Huber, Log-Cosh, R^2와 같은 대표적인 성능 지표를 활용하였으며, 각 변수의 통계적 특성과 손실함수 간의 적합성에 기반한 분석을 진행하였다. 단순히 모든 변수에 같은 손실함수를 적용하기보다, 변수의 연속성, 이상치, 분포 형태 등을 고려해 적절한 손실함수를 선택함으로써 예측 정확도와 과적합 방지의 균형을 맞추고자 하였다.

7.2.1. R^2을 통한 통합적 분석

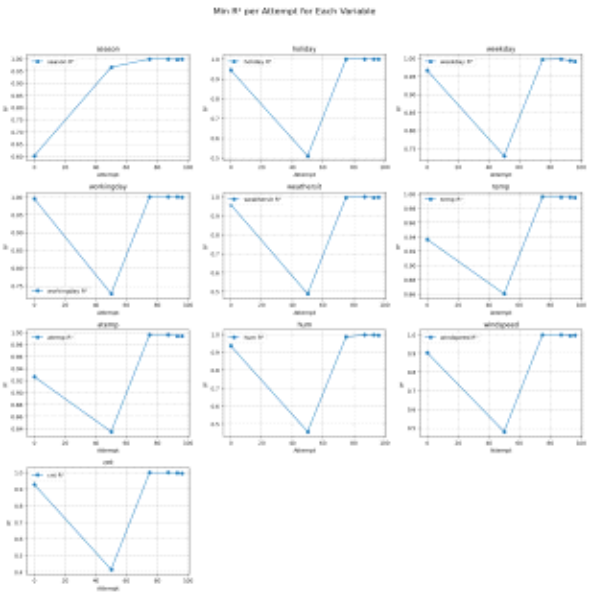

그림 38 변수별 인간 데이터 비중에 따른 R^2 최솟값.

모델의 전체적인 정확도를 나타내는 지표로써 R^2을 기초적인 해석 도구로 사용하였다. 그래프 분석 결과, season 변수를 제외한 모든 변수에서 뚜렷한 V자형 패턴이 관찰되었다. 인간 데이터의 비율이 0%일 때 대부분 변수에서 R^2 값이 0.9 이상으로 높게 시작하였으나, 50% 지점에서 급격히 하락하여 최저점을 기록하였다. 이후 75% 지점에서 모든 변수의 R^2 값이 다시 0.95 이상으로 급격히 상승하였고, 87.5%와 100% 지점에서는 1 수준으로 유지된다. 다만, weekday, attemp, hum 등의 변수에 대해서는 R^2이 소폭 하락하는 현상을 보였다.

이는 인간 데이터와 비인간 데이터가 균등하게 혼합된 상태(50%)에서 모델이 변수 간 관계를 포착하는 데 어려움을 겪지만, 어느 한쪽으로 데이터 비율이 치우칠 때(0% 또는 87% 이상) 모델의 예측 성능이 향상됨을 시사한다. 이러한 경향성은 특히 holiday, weekday, workingday 등 이산형 변수에서 큰 경향성을 보였다.

7.2.2. RMSE를 통한 연속형 데이터 분석

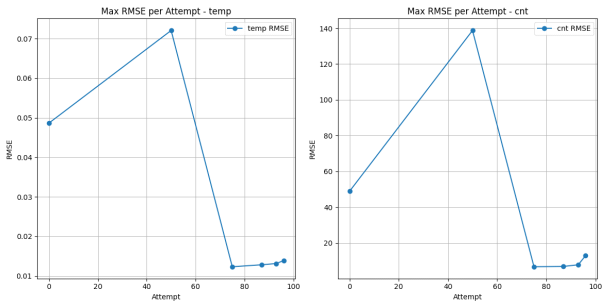

그림 39 변수별 인간 데이터 비중에 따른 RMSE 최댓값

temp, cnt와 같은 연속형 변수는 상대적으로 분산이 크며, 어느 정도의 불규칙성을 가진 변수이다. 불규칙성이 변수는 특성상 큰 오차에 민감한 RMSE 손실함수가 적합하게 작용한다. 인간 데이터 0%에서는 각각 0.05와 40 수준으로 상대적으로 높았다. 후에 50%에서 두 변수 모두 극댓값을 갖고, 인간 데이터 비율이 75% 이상으로 증가한 구간에서는 이들 손실함수에서의 오차 값이 0.2에서 0.1로 급감하며 데이터의 정확도가 향상되었음을 확인할 수 있었다. 그러나 해당 데이터 역시 인간의 데이터 비율이 87.5%에서 96%, 100%로 증가하는 경우 오차율 또한 커지는 경향성을 가졌음을 확인할 수 있다.

7.2.3. MAE를 통한 범주형 변수의 분석

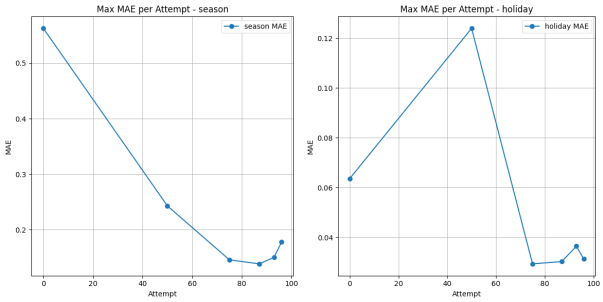

그림 40 변수별 인간 데이터 비중에 따른 MAE 최댓값

season, holiday와 같은 치역이 적은 변수들은 오차값과 정비례하는 손실함수인 MAE를 사용하는 것이 적합하다. 모두 75% 지점부터 세대에 따라 데이터의 경향성이 감소하는 모습을 확인할 수 있었다. 또한, 인간 데이터 비율이 96%, 100%인 경우, 최댓값이 증가하는 경향성을 보였다.

7.2.4. HUBER Loss를 통한 이상값 존재 함수 분석

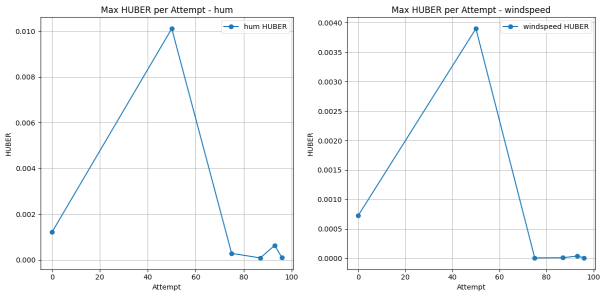

그림 41 변수별 인간 데이터 비중에 따른 HUBER Loss 최댓값

humidty, holiday 등 이상치가 존재하는 연속형 변수는 MAE와 RMSE의 경향성을 모두 포함한 HUBER Loss 함수를 이용하는 것이 이상적이다. 두 변수 모두 50%에서 극댓값을 보였으며, 이후에는 약 0.01 수준으로 매우 감소하는 경향을 띤다. 단 데이터가 87.5%에서 96%로 증가할 때, 두 함수 모두 값이 소폭 증가한다.

7.2.5. Log-Cosh 함수를 통한 주기 함수의 분석

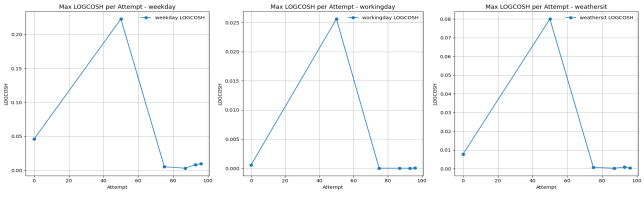

그림 42 변수별 인간 데이터 비중에 따른 Log-Cosh 최댓값

weekday, workingday, weathersit 와 같은 일정한 주기성을 띠는 변수에 대해서는 Log-Cosh 함수를 이용하여 적은 오차율에 강한 페널티, 큰 오차율에 적은 페널티를 주는 것이 유리하다. 세 변수 모두 50%에서 극대를 나타낸다. 그러나, 인간 비율이 87.5%에서 96%로 증가할 경우, weathersit 및 weekday 변수는 오차율이 소폭 상승하지만, workingday 변수에 대해서는 약 0.05 수준의 변화만 감지되었을 뿐, 주목할 만한 변화가 없었다.

7.2.6. 종합 결론

실험 결과에서 주목할 만한 현상은 인간 데이터 비율이 87.5%에서 96%, 96%에서 100%로 증가할 때 발생한 성능 저하로, 이는 전형적인 과적합의 징후로 해석된다. 특히, RMSE, MAE, R^2, HUBER, Log-Cosh 등 모든 손실함수에서 오차가 증가한 것이 이를 뒷받침한다. 이는 모델의 데이터를 완벽히 수집한 것으로 확인되어, 오히려 손실함수의 값이 증가한 것으로 확인된다.

결론적으로, 본 실험은 인간 데이터의 비율 증가만으로는 예측 성능이 완벽히 향상되지 않으며, 변수의 속성과 손실함수 간의 논리적 대응이 함께 고려될 때 안정적이고 일반화 가능한 모델이 구축된다는 점을 입증하였다.

본 실험에서 해당 비율은 약 87%에서 96% 사이였으며, 이는 본 연구자가 설정한 가설 '인간 데이터와 AI 생성 데이터의 최적 비율은 약 20%일 것으로 예상된다' 와 약 10% 정도의 차이를 보인다. 이는 기존 데이터에 내재한 패턴의 복잡성과 분산 간 특수성이 초기 가설에서 예상했던 것보다 모델 학습에 더 큰 영향을 미쳤음을 시사한다. 특히 87.5%에서 최적의 성능을 보인 점은 인간 데이터의 주도적 역할과 함께 소량의 AI 생성 데이터(약 12.5%)가 모델의 일반화 능력을 향상하는 정규화 효과를 제공한 것으로 해석된다.

이러한 결과는 단순히 인간 데이터의 양을 늘리는 것보다 데이터의 다양성과 품질이 모델 성능에 더 중요한 요소임을 입증한다. 또한, 변수별 특성에 따라 손실함수의 적절한 선택이 모델 성능 최적화에 결정적 역할을 한다는 점도 확인되었다.

본 실험을 통해 인간 데이터와 AI 생성 데이터 간의 상호보완적 관계가 존재하며, 적합 비율이 존재함이 실험적으로 확인되었다.

8. 제언

본 연구는 인간 데이터 비율과 손실함수의 조합 비율에 따라 회귀 모델의 성능에 미치는 영향을 분석하였으며, 변수의 특성과 손실함수 간의 적합성이 모델의 일반화 성능 향상에 중요한 역할을 한다는 사실을 실험적으로 입증하였다. 이러한 결과를 바탕으로 다음과 같은 제언을 제시한다.

둘째, R^2과 같은 전반적인 설명력 지표는 모델 성능의 전체적인 경향을 파악하는 데 유용하지만, 개별 손실함수의 수치만으로는 포착하기 어려운 정량적 성능 격차가 존재할 수 있다. 따라서 후속 연구에서는 R^2그럴 뿐만 아니라 손실함수별 세부 지표 분석 및 시각화를 추가로 적용한 성능 평가 체계를 확립할 필요가 있다.

셋째, 본 실험에서는 변수별 손실함수 조합의 성능 차이를 인간 데이터 비율이라는 단일 축을 중심으로 분석하였으나, 실제 응용에서는 데이터의 시간적 구조, 노이즈의 비정형성, 그리고 사용자 행동의 예측 불가능성과 같은 요소들도 복합적으로 작용한다. 따라서 향후 연구에서는 다양한 조건으로 손실함수의 유연성과 안정성을 평가하는 후속 실험이 필요하다.

마지막으로, 실제 응용 단계에서는 데이터 확보의 비용과 정확도 간의 균형에 대한 고려가 필요하다. 본 연구는 인간 데이터의 비율이 높아질수록 모델 성능이 향상된다는 점을 보였으나, 이는 현실적 관점에서 자료수집의 비용과 시간이라는 한계를 수반한다. 따라서 앞으로는 적은 양의 고품질 인간 데이터를 기반으로 성능을 유지할 수 있는 손실함수 최적화 및 학습 모델 설계 또한 중요한 연구 방향이 될 것이다.

참고문헌

[1] Alex Graves et al (2013). Speech recognition with deep recurrent neural networks, IEEE, https://ieeexplore.ieee.org/document/6638947

[1] Borji, A. (2024). A Note on Shumailov et al. (2024): 'AI Models Collapse When Trained on

Recursively Generated Data'. arXiv. https://arxiv.org/html/2410.12954v1

[2] Chen, M. (2022). Vanishing Gradient Problem in Training Neural Networks. 10.25911/BMDG-3N85.

[3] Dohmatob, E., Feng, Y., & Kempe, J. (2024). Model collapse demystified: The case of regression. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, & C. Zhang (Eds.), Advances in Neural Information Processing Systems (Vol. 37, pp. 46979-47013). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2024/file/53dbd7e34fab703a639964e2d3ee9e84-Paper-Conference.pdf

[3] Emmah, V. (2022). Performance evaluation of LSTM and RNN models in the detection of email spam messages. European Journal of Information Technologies and Computer Science, 2(6). https://doi.org/10.24018/COMPUTE.2022.2.6.80

[4] Fanaee-T, H. (2013). Bike Sharing [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5W894.

[5] Ghislieri, M., Cerone, G.L., Knaflitz, M. et al. (2021). Long short-term memory (LSTM) recurrent neural network for muscle activity detection. Journal of NeuroEngineering and Rehabilitation, 18, 153. https://doi.org/10.1186/s12984-021-00945-w
Lucas B.V. de Amorim, George D.C. Cavalcanti, Rafael M.O. Cruz, The choice of scaling technique matters for classification performance, Applied Soft Computing,
Volume 133, 2023, 109924, ISSN 1568-4946,
https://doi.org/10.1016/j.asoc.2022.109924.

[6] Lucassen, M., Decker, T., Guzmán, F.G. et al. (2023). Simulation methodology for the identification of critical operating conditions of planetary journal bearings in wind turbines. Forschung im Ingenieurwesen, 87, 147-157.

[7] Noh, S.-H. (2021). Analysis of Gradient Vanishing of RNNs and Performance Comparison. Information, 12(11), 442. https://doi.org/10.3390/info12110442

[8] Pilla, P. R., & Mekonen, R. (2025). Forecasting S&P 500 Using LSTM Models. Zenodo. https://doi.org/10.5281/zenodo.14759118

Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. Neural Comput 1997; 9 (8): 1735-1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735

[9] Shumailov, I., Shumaylov, Z., Zhao, Y. et al. (2024). AI models collapse when trained on recursively generated data. Nature, 631, 755-759. https://doi.org/10.1038/s41586-024-07566-y

[10] Tian, R. (2024). Google Stocks Prediction by Machine Learning of RNN and LSTM Techniques. Advances in Economics, Management and Political Sciences, 57, 285-293.

[11] 김민승, 이재식, 오은식, 이찬호, 최지혜, 장용주, 이정희, 성태응. (2021). 딥러닝 기반 지능형 기술가치평가에 관한 연구: 심층신경망 학습을 통한 정성평가 지표 예측 모형. 기술혁신학회지, 24(6), 1141-1162.

[12] 이승연, 허석렬, 이완직. (2024). 인공지능 재귀 학습에 따른 모델 붕괴 현상 개선 방안. 한국소프트웨어감정평가학회 논문지, 20(4), 145-154.

[13] 조준모. (2019). 빅데이터의 정규화 전처리 과정이 기계학습의 성능에 미치는 영향. 한국전자통신학회 논문지, 14(3), 547-552.

[14] Cloudflare. (2024, May 16). Declaring your AI-ndependence: Block AI bots, scrapers, and crawlers with a single click. Cloudflare. https://blog.cloudflare.com/declaring-your-aindependence-block-ai-bots-scrapers-and-crawlers-with-a-single-click/

[15] 정규화. 네이버 지식백과(AI 용어사전), https://terms.naver.com/search.naver?query=%EC%A0%95%EA%B7%9C%ED%99%94

[15] Pavel Holoborodko (n.d.). QuickLaTeX.com - LaTeX equation rendering for websites and forums. Retrieved from https://www.quicklatex.com/

LLM 대화 내역

https://chatgpt.com/share/683702a4-efa8-800a-8d27-54ef93304342
데이터 병합 알고리즘 질문, 25.05.21.

https://chatgpt.com/share/6837030d-f800-800a-9bf2-921a619bf8c2
pandas 관련 문법 질의 및 디버깅, 25.05.20

https://chatgpt.com/share/68370420-535c-800a-811e-917aa9bac023
하이퍼파라미터 튜닝, 25.05.15

https://chatgpt.com/share/68370464-589c-800a-aa86-98e7ce5a7123
CNN과 LSTM의 비교, 25.05.15

https://chatgpt.com/share/683575ff-3988-800a-9806-4b32d989ecbd
AI 붕괴의 증명 엄밀성 검증, 25.05.28.

https://chatgpt.com/share/6837050b-2b14-800a-bec4-8145488ccbf5
데이터프레임 셔플링 함수 생성, 25.05.23.

https://chatgpt.com/share/68370520-5038-800a-8f05-c89e264adfd1
하이퍼파라미터 튜닝, 25.05.16.

https://chatgpt.com/share/6837053e-bf24-800a-8126-84e6ff9e5736
선행 연구 번역, 25.05.08.

https://chatgpt.com/share/68370553-8b00-800a-bf36-44b88f08ab46
논문 양식 요약정리, 25.05.06.

https://claude.ai/share/582edbdf-22f4-4212-ad31-b425c1ec5bf9
데이터 시각화 순서 코드, 25.05.13.

https://claude.ai/share/f36b7329-6e19-4316-bdb3-cfdd0f307d34
LSTM 출력 형식 개조, 25.05.10.

https://chatgpt.com/share/683708fd-17d0-8008-aadb-eb084a280b7e
다양한 손실함수의 설명 및 식 질문

https://chatgpt.com/share/68370996-a994-8008-ba40-4017f330dfae
1번 실험에 대한 결론 및 방향성 질문

실험 코드 전문 및 결과 RAW DATA

1번 실험 결과: https://drive.google.com/drive/folders/1qyTvUJozFf8f8KknzIenq3iQSO_d57Tm?usp=drive_link

2번 실험 결과: https://drive.google.com/drive/folders/1stkyj5SAtEBrBfgDoR025FKndL64IoWQ?usp=drive_link

코드 전문: https://colab.research.google.com/drive/1I92xteJVG6M7KxUEysxMbNXa6Ap-s_v_?usp=sharing