# MCU 기반 Edge 장치에서의 온디바이스 학습을 위한 플래시 메모리 활용 연구

이세인 [1], 곽준호 [2], 전제홍 [1], 조정훈 [3]
[1] 경북대학교 대학원 전자전기공학부 석사과정
[2] 경북대학교 대학원 전자전기공학부 박사과정
[3] 경북대학교 전자공학부/대학원 전자전기공학부 교수

lsin07@knu.ac.kr, junho7513@knu.ac.kr, jehongjeon@knu.ac.kr, jcho@knu.ac.kr

# A Study on Flash Memory Utilization for On-Device AI Learning on MCU-based Edge Devices

Sein Lee[1], Junho Kwak[1], Jehong Jeon[1], Jeonghun Cho[1 2]
[1]Graduate School of Electronic and Electrical Engineering, Kyungpook National University
[2]School of Electronic Engineering, Kyungpook National University

**요 약**

최근 인공지능 기술이 다양한 분야로 확산되면서 사용자 디바이스에서 인공지능 모델을 학습시키는 온디바이스 학습 기술이 주목받고 있다. 이는 로컬 환경에서 데이터를 수집과 모델 학습을 모두 수행함으로써 개인정보의 노출 방지와 네트워크 지연 최소화를 가능케 한다. MCU 기반 Edge 디바이스에 지속적인 동작이 가능한 인공지능 모델을 탑재하기 위해서는, 저장공간 제약과 전원의 차단과 복귀가 반복적으로 발생하는 디바이스의 동작 특성을 고려한 프레임워크의 설계가 필요하다. 본 연구에서는 이와 같은 조건을 만족하도록, 모델의 연속적인 학습이 가능하며 하드웨어 이식성이 있는 비휘발성 메모리 활용 인공신경망 프레임워크를 구현하고, 실험을 통해 전원 차단 이후에도 세션이 지속적으로 유지되고 모델 검증 절차가 정상적으로 동작함을 확인하였다.

## 1. 서론

최근 인공지능 기술은 다양한 분야에서 빠르게 확산되며, 사용자 맞춤형 서비스의 필요성이 점차 강조되고 있다. 특히 사용자의 특성, 행동, 취향 등을 반영하여 개인화된 경험을 제공하기 위한 Edge AI 기술이 교육, 헬스케어, 스마트 기기 등의 영역에서 핵심적인 역할을 하고 있다. 이를 통해 일반화된 모델이 제공하지 못하는 사용자 특화 서비스를 제공함으로써 높은 사용자 만족도를 달성할 수 있으며, 사용자의 실제 환경과 밀접하게 연계된 예측 및 판단이 가능하다는 장점이 있다.

인공지능 모델의 학습과 추론을 사용자 디바이스에서 독립적으로 수행하는 온디바이스 AI 기술은 사용자 데이터를 네트워크를 통해 전송하지 않고 디바이스 내에서 학습 및 추론을 수행함으로써 개인정보의 노출을 막고 네트워크 지연의 최소화가 가능하다. 이러한 장점으로 인해, 온디바이스 AI 기술이 Edge AI 기술의 핵심 부분 중 하나로 주목받고 있다.[1]

온디바이스 AI 모델, 특히 MCU 기반으로 동작하는 모델은 기존의 풍부한 컴퓨팅 자원을 바탕으로 구동되는 모델에 비해 사용 가능한 저장공간의 용량 등 디바이스 사양 측면에서 크게 불리하다는 단점이 있다. 특히, 일반적으로 전원이 지속적으로 공급되지 않고 차단과 복귀가 지속적으로 반복되는 동작 패턴을 보일 가능성이 높은 MCU 기반 디바이스에서는, 제한된 플래시 메모리 공간을 효율적으로 이용하여 끊김 없는 인공지능 경험을 제공하는 것이 필수이다. 이러한 점을 고려하여, 본 논문에서는 MCU 기반 디바이스의 사용 특성을 고려하여 플래시 메모리의 효율적 이용으로 지속적인 학습이 가능한 인공지능 프레임워크를 제안한다.

## 2. 프레임워크 구조

본 논문에서는 모델 영역, 데이터 영역, 그리고 메타데이터 영역의 3 개 영역으로 구분한 플래시 메모리

구조를 제안한다.

모델 영역은 리스트 구조를 채택하여 인덱스를 통한 자유로운 접근이 가능하도록 구현하였다. 인덱스 번호를 통해 모델을 메인 메모리로 불러오거나, 반대로 메인 메모리의 모델을 플래시 메모리에 저장하거나, 저장된 모델을 삭제할 수 있다. 데이터 영역은 디바이스에서 수집한 학습에 사용할 사용자의 개인 데이터가 저장되는 영역이다. 데이터영역에는 사용자가 디바이스를 사용하면서 생성된 특성(feature)과 레이블(label) 쌍을 스택 자료구조를 이용하여 저장한다.

인공신경망 모델의 모든 가중치 값과 모든 학습 데이터는 동적 고정소수점 (Dynamix fixed point)[2] 양자화를 통해 8 비트 혹은 16 비트 정수 데이터로 저장된다.

$$f(x) = \min(\max(round(2^{FL} \cdot x), -2^{B-1}), 2^{B-1} - 1) \quad (1)$$

실제 값 $x$ ($x \in \mathbb{R}$)에 대해, 양자화된 값 $f(x)$는 위 수식과 같이 표현된다. $B$는 양자화 데이터의 비트 폭 (8 혹은 16)을 의미한다. 소수부 비트 길이 $FL$ (Fractional Length)는 전체 데이터의 최댓값과 최솟값을 바탕으로 하여 수식 (2)와 같이 동적으로 결정된다.

$$FL = (B - 1) - IL \quad (2)$$

$B$는 양자화 데이터의 비트 폭 (8 혹은 16)이고, $IL$은 정수부의 비트 길이로 수식 (3)와 같이 표현된다.

$$IL = \left\lceil \log_2 \max\left( |M| - \left|\frac{M}{2^B}\right|, |m| - \left|\frac{m}{2^B}\right| \right) \right\rceil \quad (3)$$

$M$은 전체 데이터의 최댓값, $m$은 전체 데이터의 최솟값을 의미한다. $|M|$과 $|m|$으로부터 일정 값을 뺌으로써, 데이터의 범위를 줄여 최댓값과 최소값에 대해 약간의 포화를 허용하는 대신 양자화 해상도를 개선하여 더 정밀한 값을 표현할 수 있도록 한다.

메타데이터는 플래시 메모리에 저장된 학습 데이터와 모델에 대한 정보와, 이들이 사용 중인 메모리 공간에 대한 정보를 포함한다. 메타데이터는 디바이스의 전원이 꺼져 메인 메모리의 정보가 모두 손실되더라도, 플래시 메모리에 저장된 모델과 학습 데이터에 대한 정보를 잃지 않도록 저장해두는 역할을 한다.

플래시 메모리는 이미 데이터가 기록된 영역에 다른 데이터를 덮어씌우는 동작이 불가능하고, 이미 데이터가 기록된 영역에 재기록하기 위해서는 섹터 전체의 데이터를 지우고 새로 기록하야 한다는 물리적인 한계가 있다.[3] 메타데이터는 다른 데이터가 변경될 때마다 지속적으로 업데이트되어야 하므로, 메타데이터에 대해서는 개별 섹터를 할당하고, 모델을 2개 이상 저장하는 경우에도 서로 다른 모델을 각각 취급이 가능하도록 각 모델에 개별 섹터를 할당해야 한다.

이와 같은 제약사항들을 모두 고려하였을 때, 플래시 메모리에 데이터가 저장되는 구조는 그림 1 과 같이 표현할 수 있다.

플래시 메모리의 주소 공간이나 섹터 크기 등은 각 하드웨어마다 상이하므로, 플래시 메모리 주소 공간 정보, 모델 또는 데이터의 크기, 그리고 모델을 저장할 섹터 번호 등을 헤더 파일 내에 컴파일러 매크로로 정의하여 전처리 과정에서 실제 주소로 알맞게 변환되도록 구현하였다. 이를 통해 코드의 하드웨어 이식성을 확보할 수 있다. 플래시 메모리 주소 공간과 마찬가지로, 플래시 메모리에 하드웨어적으로 접근하기 위한 드라이버나 하드웨어 추상화 계층 역시 각 하드웨어마다 서로 다르다. 따라서, 플래시 읽기/쓰기/지우기와 같은 하드웨어 접근을 위한 코드는 사용하려는 하드웨어에 알맞게 작성하여 프레임워크에 메소드로 등록하여 사용할 수 있도록 하였다.

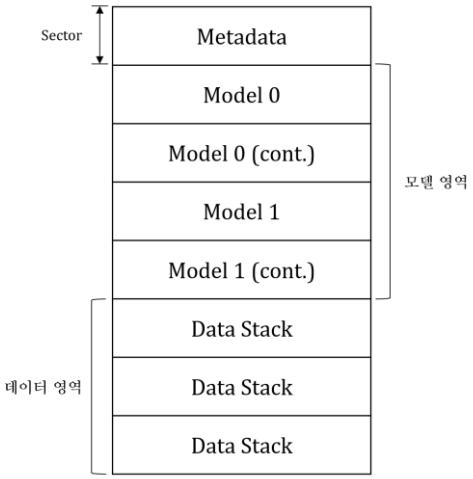


그림 1. 플래시 메모리 맵

## 3. 프레임워크 동작

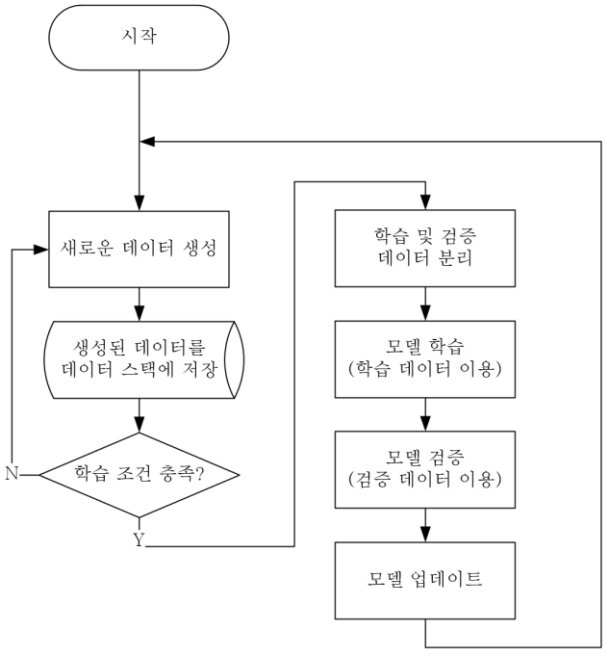


그림 2. 프레임워크 전체 동작 순서도

그림 2 는 본 논문에서 제안하는 프레임워크의 동작 순서를 나타낸 순서도이다. 디바이스의 실사용 중 학습에 사용할 수 있는 특성 및 레이블 쌍이 생성되면, 생성된 데이터를 플래시 메모리의 데이터 스택에 Push 하도록 그림 3 와 같이 주 제어 워크플로우에서 프레임워크 동작을 호출한다.

프레임워크는 미리 정해진 학습 시작 조건을 충족할 때까지 생성된 데이터의 축적을 반복한다. 학습 시작 조건의 판별은 별개의 함수로 정의하여, 데이터의 개수 혹은 데이터 수집 시작 시점으로부터 흐른 시간 등 경우에 따라 다양한 조건을 적용할 수 있도록 구현하였다.

학습 시작 조건이 충족되어 프레임워크가 학습을 시작해도 좋다고 판단하였을 경우, 프레임워크는 우선 저장된 데이터를 학습 데이터와 검증 데이터로 구분한다. 학습 데이터는 그림 4 와 같이 실제 모델 학습에 사용되고, 검증 데이터는 학습 시작 전과 후의 모델의 성능을 검증하여 모델의 성능 향상을 확인하는 용도로 사용된다.

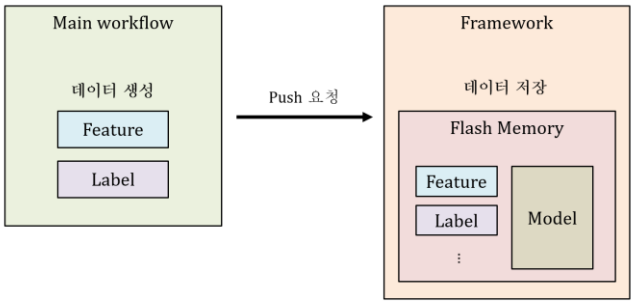


그림 3. 데이터 Push 요청

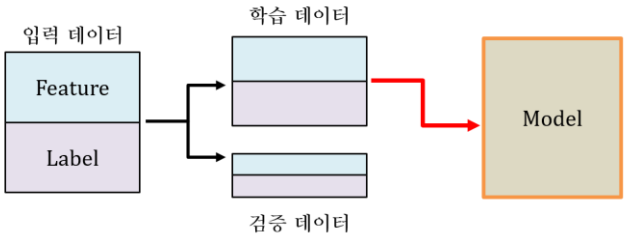


그림 4. 학습 데이터를 이용하여 모델 학습

학습이 완료되면, 학습 데이터와는 별개의 검증 데이터를 이용하여 모델을 검증하여 성능 지표를 구한다. 이 성능 지표를 바탕으로 모델을 업데이트하거나, 기존 모델을 그대로 유지할 수 있다.

## 4. 실험 결과

본 논문에서 제안하는 프레임워크 구조의 동작을 확인하기 위해서, 프레임워크를 구현하여 실제 마이크로컨트롤러 상에서 그 동작을 확인하였다. 회귀 문제를 해결하는 인공신경망 모델을 본 논문의 플래시 메모리 활용 프레임워크를 통해 학습시켰으며, 모델 학습을 위해서는 [4]에서 제안한 Edge AI 학습 플랫폼을 이용하였다. 표 1 에는 실험에 사용한 마이크로컨트롤러의 하드웨어 사양, 표 2 에는 실험을 위해 설정한 인공지능 모델의 구조, 그리고 표 3 에는 모델의 학습 관련 설정 관련 내용이 포함되어 있다.

표 1. 실험 환경

| | |
|---|---|
| Microcontroller | STM32F411 |
| Processor | Arm® Cortex®-M4 |
| Main Memory | 128 KB |
| Flash Memory | 512 KB |

표 2. 인공신경망 모델 구조

| | |
|---|---|
| 모델 구조 | 1x Input Layer, 3x Dense Layer |
| Input 크기 | 4 |
| Output 크기 | 1 |
| Dense 은닉층 크기 | 16 |
| Dense Layer 활성화 함수 | ReLU |

표 3. 모델 학습 설정

| | |
|---|---|
| 학습률 | 0.001 |
| 최적화 알고리즘 | Adam |
| 손실 함수 | MSE |
| Epochs | 50 |
| Batch 크기 | 32 |

학습에 필요한 데이터는 매 시점에 디바이스에서 생성되는 것을 사용하는 것이 원래의 개발 의도이나, 본 실험에서는 프레임워크의 정상 동작을 확인하는 것에 초점을 두어 실제 마이크로컨트롤러 애플리케이션 환경을 설정하는 것을 PC 에서 UART 통신을 통해 특성/레이블 쌍을 전송하는 것으로 대체하였다. 학습은 [5]의 데이터셋 중 ‘AT’, ‘V’, ‘AP’, ‘RH’를 입력 특성으로, ‘PE’를 출력 레이블로 사용하여 진행되었다. 그리고, 데이터 축적 과정 중 MCU 의 전원을 차단하였다가 다시 작동시켜 봄으로써 플래시 메모리의 정상적인 동작을 확인해 보았다.

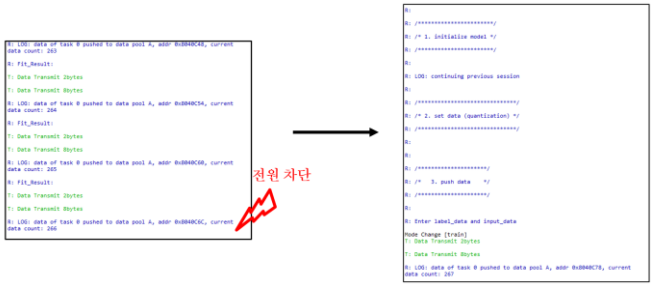


그림 5. 데이터 수집 과정에서의 전원 차단과 그 이후의 지속적 세션 수행

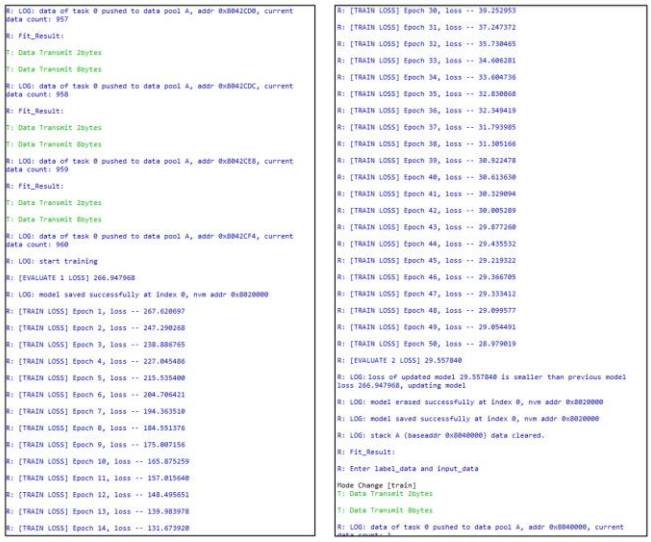


그림 6. 모델 학습 및 업데이트

그림 5 에서 보이는 바와 같이, 디바이스 사용 중 전원이 차단되더라도, 재부팅 이후 기존의 데이터 구조를 인식하여 정상적으로 다음 데이터 저장이 계속되고 있는 것을 확인할 수 있다. 그림 6 에서는, 그림 5 와 같이 데이터 수집 중 전원이 차단되었더라도 정상적으로 모델 학습이 가능하며, 모델 검증 및 업데이트 과정도 마찬가지로 정상적으로 진행되고 있음을 보여준다.

## 5. 결론

본 연구에서는 MCU 기반 디바이스의 플래시 메모리를 활용하여 사용자 데이터를 실시간으로 수집하고 디바이스 내에서 학습할 수 있는 환경을 마련하였다. 본 프레임워크의 개발을 통해, 디바이스 사용자의 개인 데이터 외부 유출의 위험 없이 사용자에게 특화된 인공지능 경험을 제공하는 디바이스 애플리케이션을 개발하기 위한 기반을 마련하였다. 또한, 하드웨어 동작을 추상화하고 플래시 메모리 구조에 대한 인터페이스를 마련함으로써, 하드웨어 이식성을 높이고 대규모의 코드 수정 없이 다양한 하드웨어에 본 프레임워크를 적용할 수 있다.

추후 우리는 데이터가 지속적으로 생성되는 환경에 알맞은 모델 학습 알고리즘을 탐색하고, 모델 평가 알고리즘을 보완하여 성능이 높은 모델을 더 잘 구별해 낼 수 있도록 지속적인 연구를 수행할 예정이다.

## ACKNOWLEDGMENT

이 논문은 2025 년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임 (RS-2024-00415938,
2024 년 산업혁신인재성장지원사업).

## 참고문헌

[1] 홍아름, 최가은. 인공지능 분야 산업 · 기술 동향 및 이슈. 한국전자통신연구원. 2025.

[2] P. M. Gysel, "Ristretto: Hardware-Oriented Approximation of Convolutional Neural Networks." Order No. 10165798, University of California, Davis, United States -- California, 2016.

[3] C. Lee, S. H. Baek and K. H. Park, "A Hybrid Flash File System Based on NOR and NAND Flash Memories for Embedded Devices," in IEEE Transactions on Computers, vol. 57, no. 7, pp. 1002-1008, July 2008.

[4] 곽준호, 전제홍, 조정훈. MCU 기반 인공지능 학습을 위한 Edge AI 개발 플랫폼 연구. 한국통신학회 학술대회논문집, 제주, 2024.

[5] P. Tfekci and H. Kaya. "Combined Cycle Power Plant," UCI Machine Learning Repository, 2014. [Online]. Available: https://doi.org/10.24432/C5002N.