MCU 기반 인공신경망 프레임워크 상에서의 온디바이스 SVM 학습 레이어 구현

이세인, 마준익, 김규민, 곽준호, 조정훈*

경북대학교, *경북대학교

lsin07@knu.ac.kr, macs6848@knu.ac.kr, kyumin747@knu.ac.kr, junho7513@knu.ac.kr, *jcho@knu.ac.kr

Implementation of On-Device SVM Training Layer on Neural Network Framework for Microcontrollers

Sein Lee, Junik Ma, Kyumin Kim, Junho Kwak, Jeonghun Cho*

Kyungpook National Univ., *Kyungpook National Univ

요 약

본 논문은 MCU의 자원 제약을 극복하고 적응형 AI를 구현하기 위해, NNoM 프레임워크 기반의 온디바이스 SVM 학습 기법을 제안한다. 메모리 효율적인 SMO 알고리즘과 다중 분류를 위한 OvO 전략을 적용한 SVM 레이어를 설계하고, 입력 레이어를 이용한 데이터 버퍼링을 통한 학습 파이프라인을 구축하였다. 이를 통해 저사양 기기에서도 효율적인 독립 학습과 추론이 가능함을 입증하였다.

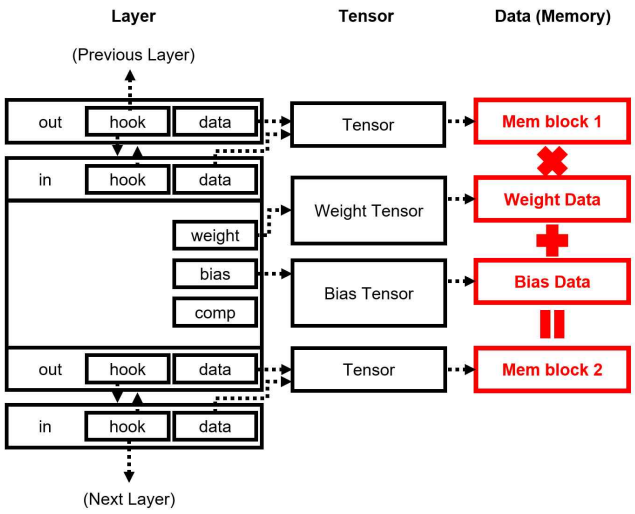
Ⅰ. 서 론

최근 사물인터넷(IoT)과 웨어러블 디바이스의 보급이 확산됨에 따라, 데이터를 클라우드로 전송하지 않고 에지(Edge) 단말에서 직접 처리하기 위한 TinyML 기술이 주목받고 있다. 특히, 인공신경망 구조를 활용하는 딥러닝(Deep Learning) 기술의 발전은 엣지 디바이스 상에서도 이미지 분류나 음성 인식 등의 복잡한 작업 수행이 가능하다는 것을 보여주었다. 그러나, 일반적인 심층 신경망(DNN)은 수백 KB 수준의 SRAM과 낮은 연산 능력을 가진 마이크로컨트롤러(MCU) 환경에서 구동하기에는 여전히 과도한 메모리 사용량과 연산 비용을 요구한다.[1] 양자화(Quantization) 및 가지치기(Pruning) 등의 모델 경량화 기법을 통해 추론 단계의 연산량은 줄일 수 있었으나, 역전파(Backpropagation)와 기울기(Gradient) 저장이 필요한 학습 과정을 MCU 내부에서 수행하는 것에는 하드웨어 자원의 물리적 한계로 인한 많은 어려움이 따른다.

MCU의 제한된 연산 능력과 메모리 공간 내에서 온디바이스 학습을 구현하기 위한 대안으로 본 논문에서는 서포트 벡터 머신 (Support Vector Machine, SVM)을 적용하였다. SVM은 Convex Optimization에 기반하므로, 적은 양의 데이터로도 안정적인 일반화 성능을 보인다.[2] 이는 데이터 수집하고 축적할 역량이 부족한 임베디드 환경에 매우 적합한 특성이다. 하지만, ARM의 CMSIS-DSP와 같은 현재 임베디드 개발 환경에서 널리 사용되는 라이브러리들은 추론 함수만을 제공할 뿐, 새로운 데이터에 대해 모델을 학습시킬 수 있는 알고리즘은 제공하지 않는다.[3] 이러한 문제를 해결하기 위해, 본 논문에서는 제한된 메모리와 연산 능력 하에서도 동작 가능한 임베디드 기반 SVM 학습 기법을 제안한다.

Ⅱ. 본 론

본 학습 프레임워크는 임베디드용 경량 인공신경망 프레임워크인 NNoM[4]과 이를 확장한 [5]의 연구에서 제안된 아키텍처를 기반으로 한다. NNoM은 C 언어로 작성된 객체 지향적 구조를 따르며, 모델을 구성하는 입력, 출력, 그리고 다수의 연산 레이어들을 독립적인 객체로 관리한다. 이 프레임워크에서 모든 레이어는 공통된 인터페이스를 통해 제어되며, 각 레이어의 동작은 레이어 연산 중간값과 최종 결과를 저장할 메모리 공간을 할당하는 Build, 이전 레이어로부터 전달 받은 입력 데이터를 바탕으로 레이어 연산을 수행하여 출력 값을 계산하는 Run, 그리고 손실 함수로부터 계산된 오차 또는 최적화 알고리즘에 따라 내부 파라미터를 갱신하는 Train이라는 3개의 핵심 동작 함수에 의해 정의된다.



〈그림 1〉 NNoM 레이어의 Run 동작

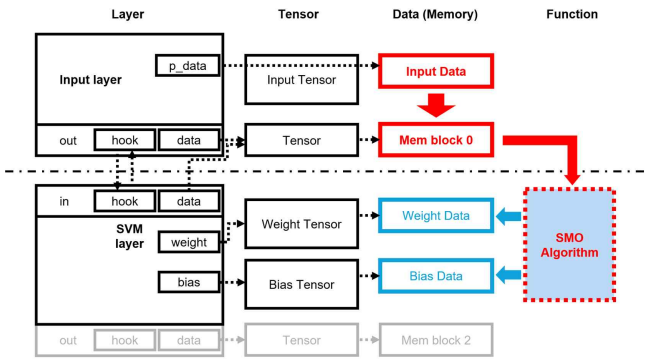
본 논문에서는 이러한 NNoM의 표준 레이어 구조 하에서, 기존의 딥러닝 레이어 대신 SVM 알고리즘을 수행할 수 있는 SVM 레이어를 설계하였다. SVM 레이어는 데이터 분류를 위한 결정 초평면(Decision

Hyperplane)을 정의하기 위해 가중치 벡터($\mathbf{w}$)와 편향(b)을 핵심 파라미터로 보유한다. 이 파라미터들은 레이어의 weight와 bias 텐서 공간에 저장된다. 추론을 수행하는 Run 과정에서는, 입력 샘플 벡터 $\mathbf{x}$에 대해서 결정 초평면을 기준으로, 아래 수식 (1)의 결과 부호에 따라 클래스를 판별한다.

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b) \quad (1)$$

Train 단계에서는 MCU의 제한된 메모리 용량을 고려하여 SMO(Sequential Minimal Optimization)[6] 알고리즘을 사용하여, 초평면의 마진(Margin)을 최대화하는 방향으로 $\mathbf{w}$와 b를 수렴시키도록 구현하였다. 이러한 방식은 대규모의 행렬 연산을 필요로 하지 않으므로, 메모리 제약이 심한 임베디드 환경에서의 온디바이스 학습에 적합하다.[7]

기존 프레임워크의 구조 하에서 SVM을 학습시키기 위해, 본 연구에서는 입력 레이어를 입력 버퍼로 사용하여 SVM 레이어에 학습할 데이터를 전달하도록 설계하였다. 추론 모드에서는 단일 샘플만을 처리하지만, 학습 모드에서는 학습에 사용할 데이터를 모두 참조해야 하므로 입력 텐서 메모리 공간을 한 번의 학습에 사용할 데이터 수만큼 저장할 수 있도록 입력 텐서 공간을 설정한다. 학습 데이터가 수집될 때마다, 이 입력 텐서를 입력 버퍼로 활용하여 순차적으로 데이터를 텐서에 적재한다. 데이터가 축적되어 입력 텐서가 가득 차면 입력 레이어의 Run 함수를 호출한다. 즉, <그림 2>에 나타난 것처럼 축적된 입력 데이터가 NNoM의 순전파 경로를 활용하여 SVM 레이어로 전달된다. 데이터를 전달받은 SVM 레이어는 추론 모드에서와 달리 Train 함수를 내부적으로 호출하여, 레이어의 weight와 bias 텐서 공간에 저장된 각 분류기들에 대해 SMO 알고리즘을 수행한다. 각 분류기의 $\mathbf{w}$와 b는 이 단계에서 업데이트되며, 업데이트가 완료되면 입력 텐서를 비우고 다음 데이터 묶음의 입력을 기다린다.



〈그림 2〉 SVM 레이어의 학습 과정

Ⅲ. 결 론

본 논문에서는 기존 딥러닝 프레임워크의 구조를 확장하여 학습 가능한 SVM 레이어를 설계하였으며, SMO 알고리즘을 적용하여 자원이 극도로 제한된 MCU 환경에서 SVM 모델의 추론과 학습을 수행할 수 있는 기법을 제안하였다. 향후, 임베디드 환경에서의 정적 메모리 할당의 한계로 효율적으로 사용되고 있지 않는 연산 단계 간의 유휴 메모리 공간을 동적으로 재사용할 수 있도록 메모리 풀의 구조를 개선하여 메모리 사용량을 최적화하는 연구를 진행할 계획이다.

ACKNOWLEDGMENT

이 논문은 2025년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원(RS-2024-00415938, 2024년 산업혁신인재성장지원사업) 및 2025년도 교육부와 대구광역시의 재원으로 대구RISE센터가 지원하는 지역혁신중심 대학지원체계(RISE, 2025-RISE-03-001) 사업의 일환으로 경북대학교에서 수행된 연구 결과입니다.

참 고 문 헌

[1] LIN, Ji, et al. Mcunet: Tiny deep learning on iot devices. Advances in neural information processing systems, 2020, 33: 11711-11722.

[2] CORTES, Corinna; VAPNIK, Vladimir. Support-vector networks. Machine learning, 1995, 20.3: 273-297.

[3] ARM Limited. (2022). "CMSIS-DSP Library Documentation.", https://arm-software.github.io/CMSIS-DSP/main/

[4] Jianjia Ma, “A higher-level Neural Network library on Microcontrollers (NNoM)”. Zenodo, 10 30, 2020. doi: 10.5281/zenodo.4158710.

[5] 곽준호, 전제홍, 조정훈. MCU 기반 인공지능 학습을 위한 Edge AI 개발 플랫폼 연구. 한국통신학회학술대회논문집, 제주, 2024.

[5] Daemen, J., and Rijmen, V. “AES Proposal: Rijndael, Version2.,” Submission to NIST, March 1999.

[6] PLATT, John. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[7] D. Anguita, A. Ghio, S. Pischiutta and S. Ridella, "A Hardware-friendly Support Vector Machine for Embedded Automotive Applications," 2007 International Joint Conference on Neural Networks, Orlando, FL, USA, 2007, pp. 1360-1364, doi: 10.1109/IJCNN.2007.4371156.