MCU 기반 Edge 장치에서의 On-device Learning을 위한 DNN 학습 과정에 대한 연구

이세인, 곽준호, 조정훈*

경북대학교 전자공학부

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

A Study on DNN Training Process for On-device Learning on MCU-Based Edge Devices

Lee Sein, Kwak Junho, Cho Jeonghun*

School of Electronics Engineering, Kyungpook National University

요 약

본 논문에서는 제한된 자원을 가진 MCU 상에서 사용할 수 있도록 메모리 사용을 최적화한 DNN(Deep Neural Network) 모델 기반 인공지능 학습 알고리즘을 구현하고, 구성한 데이터 구조 및 알고리즘을 바탕으로 실제 MCU 환경에서 그 동작을 검증하였다. 이는 Edge 디바이스에서 추가적인 하드웨어 없이 비용 및 전력 효율적인 인공지능 모델의 학습이 가능함을 시사한다.

Ⅰ. 서론

최근 인공지능 및 사물인터넷 기술의 발전과 함께 Edge 환경, 즉 최종 사용자의 로컬 디바이스에서 인공지능 연산을 수행하는 Edge AI 기술이 주목받고 있다. 일반적으로, 인공지능 모델을 학습시키는 과정이 학습된 모델을 통해 결과를 추론하는 과정보다 훨씬 더 많은 연산량을 요구한다. 그래서, 인공지능 모델의 학습이 필요한 Edge 환경의 경우 NVIDIA Tegra와 같은 SBC(Single-Board Computer)용 고성능 SoC 혹은 AP를 이용하는 경우가 많다. 하지만, 이러한 고성능 처리기를 모든 Edge 디바이스에 도입하는 것은, 비용에 민감한 프로젝트에는 적합하지 않은 방법이다. 만약 개별 Edge 디바이스의 마이크로컨트롤러(MCU)를 이용한다면, 추가적인 하드웨어 비용 없이 적은 전력 소모로 모델의 학습을 진행할 수 있을 것이다. 하지만, 일반적으로 MCU는 물리적, 그리고 비용적 한계로 인해 그 성능과 가용 메모리 공간이 일반적인 SBC용 고성능 AP에 비해 큰 폭으로 제한되어 있다.[1] 본 논문에서는 이처럼 제한된 자원을 효과적으로 사용하기 위해, 학습 알고리즘에서의 메모리 사용량을 최적화하여, 비용 및 전력 측면에서 효율적인 MCU 기반의 인공신경망 학습 환경을 제안한다.

Ⅱ. 본론

본 논문에서는 가장 기본적인 인공신경망 모델 중 하나인 DNN (Deep Neural Network)를 기준으로 실험을 진행한다. 일반적으로 DNN의 모델 학습은 아래와 같은 일련의 역전파 과정을 통해 이루어진다.[2]

$$\begin{aligned}\delta^L &\equiv \nabla_{z^L} C = \nabla_a C \odot f'(z^L)\\ \delta^l &\equiv \nabla_{z^l} C = \left((w^{l+1})^T \delta^{l+1}\right) \odot f'(z^l)\\ \frac{\partial C}{\partial b^l_j} &= \delta^l_j\\ \frac{\partial C}{\partial w^l_{jk}} &= a^{l-1}_k \delta^l_j\end{aligned} \tag{1}$$

수식 (1)으로 표현된 모델은 대문자 L개의 층(layer)을 가지고 있으며, 각 층은 소문자 l로 표시된다. z^l, a^l 은 모델 순전파 과정에서 결정되는, 각 신경망 층 l의 출력값이고, w^l, b^l은 각 층의 매개변수들로 각각 가중치(weight)와 편향(bias)을 의미한다. 이 변수들의 관계는 수식 (2)와 같이 정의된다.

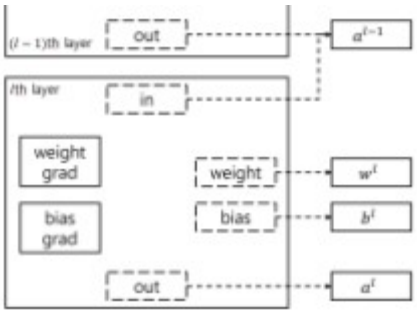
$$\begin{aligned}z^l &= w^l a^{l-1} + b^l\\ a^l &= f(z^l)\end{aligned} \tag{2}$$

f는 활성화 함수(activation function)를 의미한다. 그리고 C는 비용 함수(cost function)를, δ^l은 각 층에서의 오차값을 의미한다. 마지막으로, 수식 (3)과 같이 경사하강법(gradient descent)을 통해 가중치와 편향 파라미터를 업데이트하는 것으로 학습이 진행된다.

$$\begin{aligned}w &\leftarrow w - \eta \frac{\partial C}{\partial w}\\ b &\leftarrow b - \eta \frac{\partial C}{\partial b}\end{aligned} \tag{3}$$

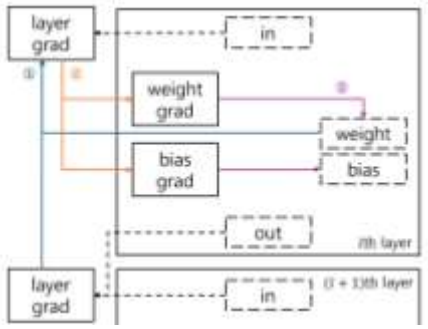
왼쪽 화살표(←)는 화살표 좌변의 값을 우변의 값으로 업데이트하라는 의미이고, η는 학습률(learning rate)을 의미한다.

학습 알고리즘과 자료구조의 구현을 체계화하기 위해, DNN의 각 신경망 층은 아래 <그림 1>과 같이 구조화된다. <그림 1>에서 점선으로 표시된 도형은 참조 관계, 실선으로 표시된 도형은 실제 데이터를 저장하고 있는 영역을 의미한다. 모델의 학습을 진행하기 위해서는 순전파 과정에서 계산된 이전 층의 출력값, 즉 현재 층의 입력값을 기억하고 있을 필요가 있다. 이전 층의 출력값은 현재 층의 입력값과 동일한 값을 나타내므로 같은 메모리 공간을 사용할 수 있다.



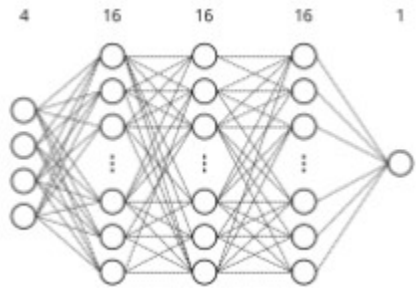
〈그림 1〉 dense layer 자료구조의 매개변수와 출력값

모델 학습 과정의 순전파 단계에서 기억되었던 각 층의 입력값은 역전파 과정에서 가중치 및 편향의 그라디언트 계산에 사용된다. 신경망 층 l에서의 역전파 과정은 <그림 2>와 같이 layer 자료구조에 저장된 가중치 데이터와, $(l+1)$ 층에서의 활성화 함수를 지나지 않은 입력값 z^l, $(l+1)$ 층에서의 그라디언트 δ^{l+1}, 그리고 별도로 구현된 활성화 함수 미분체 자료구조를 이용하여 수식 (1)의 2번째 등식과 같이 δ^l을 구한 다음, 이를 이용하여 매개변수 업데이트에 필요한 가중치 및 편향의 그라디언트를 구하는 과정으로 진행된다. 계산된 신경망 층의 그라디언트 δ^l은 이전 층 $(l-1)$에서 그대로 사용되므로 현재 층의 그라디언트 출력과 이전 층의 그라디언트 입력은 동일한 메모리 공간을 사용할 수 있다.



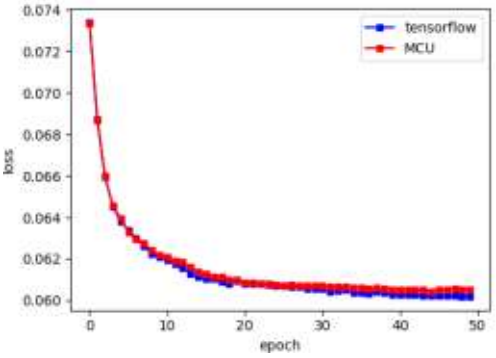
〈그림 2〉 dense layer 자료구조의 그라디언트와 역전파 과정

MCU 상 DNN 모델 학습의 검증은 ARM Cortex-M4 프로세서를 사용하는 STMiocroelectronics의 NUCLEO-F411RE 개발 보드를 통해 진행하였다. 학습을 위한 입력 데이터는 직렬 통신을 이용하여 하나의 배치(batch) 단위로 PC에서 개발 보드로 전송한다. 모델은 UCI ML Repository의 CCPP(Combined Cycle Power Plant) 데이터셋을 이용하여 학습되었으며, DNN 모델의 구조는 <그림 3>과 같이 3개의 은닉층(hidden layer)을 포함하여 구성하였다.



〈그림 3〉 CCPP 데이터셋 학습을 위한 DNN 모델 구성

실험은 PC 환경에서 tensorflow 라이브러리를 이용하여 구성한 모델과 MCU 상 학습을 위해 자체적으로 구현한 동일한 구조의 모델의 동작을 비교하여, 모델의 학습이 정상적으로 이루어지고 있는가를 손실(loss) 값을 통한 모델의 학습 성능 검증을 통해 비교하는 방식으로 이루어진다. 실험 결과는 <그림 4>와 같다.



〈그림 4〉 Tensorflow를 이용해 학습한 모델과의 학습 성능 비교

<그림 4>를 보면, MCU 상에서 학습을 진행하고 있는 모델의 에포크(Epoch)에 따른 손실값이 tensorflow를 이용한 모델의 손실값과 거의 유사한 형태로 감소하고 있음을 알 수 있다.

Ⅲ. 결론

본 연구에서는 제한된 컴퓨팅 자원을 가진 MCU에 적용이 가능한 DNN 모델 기반 인공지능 학습 알고리즘을 구현 및 검증하였다. 연구 결과, tensorflow 라이브러리를 이용하여 구성한 모델과 MCU 상에서 구성된 모델의 학습 성능이 거의 유사하다는 것을 확인하였다. 이는 Edge 환경에서 추가적인 고성능 하드웨어 없이도 비용 및 전력 효율적인 사용자 특화 인공지능 모델 학습 환경을 구현할 수 있음을 의미한다.

하지만, MCU에서의 신경망 모델 학습은 연산 성능의 한계로 인해 일반적인 PC나 SBC 환경에 비해 속도 면에서 크게 불리할 수밖에 없다. 본 논문에서 살펴본 간단한 DNN 이외의 다양한 인공신경망 모델을 MCU 상에서 유의미한 시간 내에 학습을 진행할 수 있도록 학습 구조를 개선하는 과정이 필요할 것으로 보인다.

ACKNOWLEDGMENT

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 1711160343, 차량 ECU 응용소프트웨어 개발 및 검증자동화를 위한 가상 ECU기반 차량레벨 통합 시뮬레이션 기술개발).

참 고 문 헌

[1] Marantonio Caprolu; Roberto Di Pietro; Flavio Lombardi; Simone Raponi, “Edge Computing Perspectives: Architectures, Technologies, and Open Security Issues”, 2019 IEEE International Conference of Edge Computing, pp. 116-123, Aug. 2019.

[2] Nielsen, Michael A. “How the backpropagation algorithm works”, Neural Networks and Deep Learning, Determination Press, 2015.