

CNN에서 입력 최댓값을 이용한 SoftMax 연산 기법

SoftMax Computation in CNN Using Input Maximum Value

Hyeong-Ju Kang*

*Associate Professor, School of Computer Science and Engineering, Korea University of Technology and Education, Cheonan, 31253 Korea

ABSTRACT

A convolutional neural network(CNN) is widely used in the computer vision tasks, but its computing power requirement needs a design of a special circuit. Most of the computations in a CNN can be implemented efficiently in a digital circuit, but the SoftMax layer has operations unsuitable for circuit implementation, which are exponential and logarithmic functions. This paper proposes a new method to integrate the exponential and logarithmic tables of the conventional circuits into a single table. The proposed structure accesses a look-up table (LUT) only with a few maximum values, and the LUT has the result value directly. Our proposed method significantly reduces the space complexity of the SoftMax layer circuit implementation. But our resulting circuit is comparable to the original baseline with small degradation in precision.

Keywords : CNN, SoftMax, Accelerator, Exponential function

I. 서 론

합성곱 신경망(Convolutional neural network, CNN)은 최근에 영상 분류나 객체 검출과 같은 컴퓨터 비전 분야에서 높은 성능을 보여주며 널리 사용되고 있다 [1]-[6]. 이러한 높은 성능에도 불구하고 CNN은 많은 양의 계수와 계산을 요구하므로 임베디드 시스템에 적용

하기 어렵다는 단점을 가지고 있다. 이러한 단점을 극복하는 방법 중에 CNN을 위한 전용 회로를 사용하는 방법이 많이 연구되고 있다.

CNN의 연산들은 대부분 전용 회로로 구현하기에 적합하나, 일부 연산의 경우 이에 적합하지 않다. 특히, SoftMax 층의 경우 지수 연산이나 나눗셈 연산을 이용하는 등, 이들은 모두 효율적으로 처리하는 회로를 개발하기 어려운 연산들이다. 최근에는 나눗셈 연산을 피하려고 로그 연산을 도입하는 방법이 제안되었으나, 로그 연산 또한 회로에서 구현하기 쉽지 않다.

기존에는 지수 연산과 로그 연산을 Look-up table (LUT)로 구현하는 것이 일반적이어서 SoftMax를 구현하기 위해 여러 개의 LUT를 사용하였고[7] 이로 인해 회로의 복잡도가 증가하였다. 이 논문에서는 SoftMax 연산을 하나의 LUT만으로 구현하는 방법을 제안한다. 하나의 LUT로 구현하기 위해서는 입력값들을 모아서 LUT의 입력으로 넣어야 하는데, 이 경우 LUT 엔트리의 개수가 너무 많아지는 단점이 생긴다. 이를 극복하기 위해 입력 데이터 중 가장 큰 값들을 K개 고르고, 이 값들에서 상위 w 비트만을 취하여 LUT에 입력한다. 입력값 일부만을 사용하므로 약간의 오차가 발생하나 지수 연산과 로그 연산과 같은 복잡한 연산들이 없어지고 LUT의 개수가 한 개로 줄어서 회로의 면적을 크게 줄일 수 있다. 본 논문은 학습이 완료된 CNN에 대해 예측 과정을 수행하는 회로를 목표로 한다.

이 논문은 다음과 같이 구성되어 있다. 2장에서는 기존의 SoftMax 층 구현 방법을 소개하고 3장에서는 제안하는 방식을 설명한다. 4장에서 실험 결과를 보이고 5장에서 결론을 낸다.

II. SoftMax 층의 회로 구현

2.1. SoftMax 층의 연산

영상인식에서 CNN은 영상 분류나 객체 검출 등에 많

Received 21 December 2021, Revised 24 December 2021, Accepted 22 January 2022

* Corresponding Author Hyeong-Ju Kang(E-mail:hjkang@koreatech.ac.kr, Tel:+82-41-560-1420)

Associate Professor, School of Computer Science and Engineering, Korea University of Technology and Education, Cheonan, 31253 Korea

Open Access <http://doi.org/10.6109/jkiice.2022.26.2.325>

print ISSN: 2234-4772 online ISSN: 2288-4165

이 사용되며 이때 중요한 작업 중의 하나가 입력의 어떤 부분이 어떤 종류의 객체인지를 구별하는 것이다. 이 경우 주로 사용되는 방법이 주어진 N개 클래스 각각에 해당 확률을 구한 뒤에 이들 중 가장 큰 확률값에 해당하는 클래스로 분류하는 것이다. 이때 CNN의 출력에 보통 SoftMax 연산을 수행하는데, N개의 입력값 x_n ($0 \leq n < N$)에 대해 가장 큰 값을 명확히 구별하면서 확률의 특성을 가지도록 만드는 역할을 한다. 확률은 값의 범위가 0에서 1 사이여야 하고, 전체 합이 1이 되어야 한다. 이렇게 만드는 방법은 여러 가지가 있을 수 있으나 SoftMax에서는 N개의 출력값 z_n 을 아래 식(1)과 같이 계산한다.

$$z_n = \frac{e^{x_n}}{\sum_{j=0}^{N-1} e^{x_j}} \quad (1)$$

이와 같은 연산을 하면 결과값인 z_n 은 0에서 1 사이의 값을 가지며 z_n 들의 합은 1이 된다. 이렇게 출력된 z_n 들 중 가장 큰 값의 첨자 \max 가 CNN의 입력이 해당하는 클래스가 되며 z_{\max} 가 그 클래스에 속할 확률이 된다. 모든 출력값 z_n 을 계산할 수도 있으나, 실제로 의미가 있는 것은 입력값 x_n 중 가장 큰 값인 x_{\max} 에 해당하는 출력값 z_{\max} 이므로 이 논문에서는 z_{\max} 만을 계산하는 것을 가정한다.

2.2. SoftMax 층의 회로 구현

SoftMax 층에서 사용되는 지수 연산과 나눗셈은 모두 회로로 구현하기에 적합하지 않은 연산들이다. 이 연산들을 회로로 구현할 때는 보통 아래와 같이 지수 연산과 로그 연산으로 변환한 뒤, 각 연산을 LUT로 만들어 구현한다[7].

$$\log z_n = \log \frac{e^{x_n}}{\sum_{j=0}^{N-1} e^{x_j}} = x_n - \log \sum_{j=0}^{N-1} e^{x_j} \quad (2)$$

위와 같이 연산한 결과에 다시 지수 연산을 취하여 마지막 결과인 z_n 을 계산한다. 이러한 과정을 LUT로 구현하면 그림 1과 같다. 그림에서 EXP와 LOG는 지수 연산과 로그 연산을 수행하는 LUT들이며, MAX는 최댓값을, SUM은 합을 구하는 회로이다. 입력값들에 대해 식(2)의 마지막 항과 같이 지수 연산을 수행하고 합을

구한 뒤 로그 연산을 수행한다. 그 결과를 입력의 최댓값에서 뺀 뒤 지수를 구하면 z_{\max} 를 계산할 수 있다.

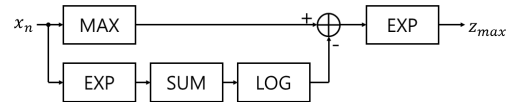


Fig. 1 Conventional SoftMax circuit

그림 1 회로에서 EXP와 LOG는 일반적으로 LUT를 이용하여 구현한다. 그러므로 그림 1 회로는 세 개의 LUT를 사용한다. 이러한 LUT의 개수를 줄이기 위해 EXP를 순차적으로 계산하는 방법도 제안되었다.[8] 이 방법에서는 식(3)과 (4)를 이용한다. 입력값의 데이터 비트 수가 w 이고 그 중 소수 부분의 비트 수가 F 일 때, 입력값을 비트 단위로 분해하면 2^{i-F} 들의 합이 되므로, 입력값의 지수승은 $\gamma_{i-F} = e^{2^{i-F}}$ 의 곱으로 표현할 수 있다. 그러므로 입력값의 i 번 비트를 순차적으로 검사하여 그 비트가 1일 때 γ_{i-F} 을 곱해 나간다. 모든 비트에 대해 이 과정을 반복하면 입력값의 지수승이 출력된다.

$$x = \sum_{i=0}^{w-1} a_i 2^{i-F} \quad (3)$$

$$e^x = e^{\sum_{i=0}^{w-1} a_i 2^{i-F}} = \prod_{i=0}^{w-1} e^{a_i 2^{i-F}} = \prod_{i=0}^{w-1} \gamma_{i-F}^{a_i} \quad (4)$$

그림 2가 순차적으로 지수 연산을 수행하는 회로이다. 기존의 큰 LUT를 대신하여 γ_{i-F} 를 저장하는 작은 LUT와 곱셈기로 구성된다.

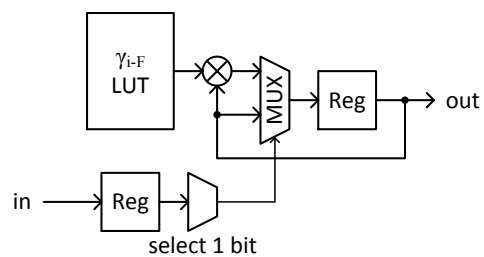


Fig. 2 Iterative exponential circuit

III. 제안하는 회로

기존의 회로에서는 LUT를 3개나 사용해야 하며, 특

히 로그 연산을 구현하는 LUT는 입력이 지수 연산의 결과이므로 비트 폭이 커서 매우 많은 엔트리가 필요하다는 단점이 있다. 이 논문에서는 이를 극복하기 위해 LUT를 한 개만 이용하는 구조를 제안한다.

영상 인식에서 CNN은 영상 분류나 객체 검출 등에 많이 이용되며, 이 경우 검출하는 객체가 속하는 클래스에 대한 $x_n(x_{max})$ 을 제외한 나머지 값들은 대부분 작은 값을 가진다. 식(1)에서와 같이 입력값들은 지수 연산 과정을 거치는데 이때 대부분의 e^{x_n} 들은 $e^{x_{max}}$ 에 비해 아주 작은 값이 된다. 즉 식(1)의 분모에 있는 합에서 항들이 대부분 작은 값이므로 x_{max} 와 비슷한 정도의 값을 가지는 x_n 들의 합으로 근사할 수 있다. 예를 들어 x_{max} 와 비슷한 크기의 입력이 x_{max1}, x_{max2} 두 개였고 나머지는 작은 값이었다면 식(1)에서 z_{max} 는 아래와 같이 근사할 수 있다.

$$z_{max} = \frac{e^{x_{max}}}{\sum_{j=0}^{N-1} e^{x_j}} \simeq \frac{e^{x_{max}}}{e^{x_{max}} + e^{x_{max1}} + e^{x_{max2}}} \quad (5)$$

식(5)에서 x_{max1} 과 x_{max2} 가 x_{max} 와 거의 비슷한 값이라면 z_{max} 는 대략 0.33 정도의 확률을 가질 것이다. z_{max} 는 N 개의 클래스 중 가장 높은 확률에 해당하므로 보통 0.3 이상의 값을 가진다. 그러므로 식(5)와 같이 x_{max} 외에 두 개 정도의 큰 값만으로도 z_{max} 를 계산할 수 있다.

제안하는 SoftMax 회로에서는 입력값 중 K 개의 제일 큰 값들을 골라서 LUT에 입력으로 넣으며, LUT는 이들로부터 z_{max} 을 계산한 값들을 직접 가지고 있다. 이렇게 하면 그림 1에서의 여러 LUT가 하나의 LUT로 통합되고, 지수 연산이나 로그 연산과 같이 회로로 만들기 어려운 부분이 사라지며, 회로 구현에 적합한 K 개의 최댓값을 고르는 동작과 LUT 접근만으로 회로를 만들 수 있다.

K 개의 제일 큰 값들을 선택한 뒤 LUT에 넣기 전에 최댓값에서 빼 주면, 식(6)에 따라 LUT의 입력으로 사용되는 값의 개수를 1개 줄일 수 있다. 또한 LUT의 엔트리 수를 줄이기 위해 각 입력값에서 상위 w 비트만을 취하여 사용하며 $x_{max_k} - x_{max}$ 가 항상 음수이므로 부호 비트를 제외하고 이용할 수 있다. 이와 같은 과정을 통해 LUT의 엔트리는 $2^{w(K-1)}$ 개가 된다.

$$z_{max} = \frac{e^{x_{max}}}{\sum_{j=0}^{N-1} e^{x_j}} = \frac{1}{\sum_{j=0}^{N-1} e^{x_j - x_{max}}} \quad (6)$$

$$\simeq \frac{1}{1 + \sum_{k=1}^{K-1} e^{x_{max_k} - x_{max}}}$$

위 식에서 max_k 는 K 개 가장 큰 값들의 첨자들이며 최댓값의 첨자는 max 이다. 제안하는 기법을 회로로 구현하면 그림 3과 같다. 입력값들로부터 K 개의 최댓값을 골라낸 뒤, 제일 큰 값과 나머지 각 값 사이의 차이를 구한다. 이 차이값들에서 부호 비트를 제외한 상위 w 비트를 추출하여 LUT에 입력하면 LUT의 출력이 z_{max} 이다.

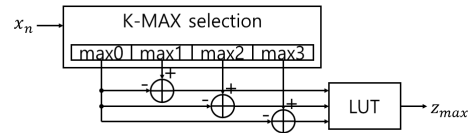


Fig. 3 Proposed SoftMax circuit

IV. 실험 결과

본 논문에서 제안하는 방법을 기존 방법과 두 가지 측면에서 비교하였다. 계산의 정확성을 비교하기 위해 부동 소수점으로 계산한 출력값과의 RMS (root-mean-square) 에러를 측정하여 비교하였고, 구현했을 경우의 복잡도를 측정하였다. 첫 번째 방식 Conventional1은 지수 연산과 로그 연산을 모두 LUT로 구현한 것이고, Conventional2는 지수 연산을 순차적으로 연산하는 것이고, 마지막으로 Proposed는 제안한 방법과 같이 최대 값만을 이용한 것이다. 입력값의 폭은 8비트이며, 이 중 1비트는 부호 비트, 3비트는 정수 부분, 4비트는 소수 부분이다. 지수승 결과의 폭은 16비트로 하였다. 제안한 방법에서는 4개의 큰 값들을 선택하였으며, 상위 4개 비트를 이용하여 LUT를 접근하였다.

계산의 정확성을 비교하기 위해 입력으로는 VOC 데이터셋을 이용한 객체 검출 상황을 가정하여 21개의 값을 임의로 생성하여 넣었으며, 1개의 값은 큰 값, 나머지는 작은 값이 되도록 생성하였다. 1000 번 생성된 데이터에 대해 부동 소수점으로 계산한 경우와의 RMS 에러를 측정한 것이 표 1이다. 기존의 방식에서는 RMS 에러가 0.027과 0.077이었으며, 제안한 방법에서는 0.051이

었다. 기존의 방법에 비해 RMS 에러가 증가하였으나, 그 차이는 크지 않았다.

Table. 1 RMS error

	Conventional1	Conventional2	Proposed
RMS Error	0.027	0.077	0.051

표 2에서는 각 방식으로 SoftMax 층 연산 회로를 구현한 뒤 GlobaFoundry 65nm 공정 라이브러리로 합성하여 EXP 블록, LOG 블록, 그리고 전체 면적을 비교하였다. 제안하는 구조에서는 EXP와 LOG가 구분되지 않고 하나의 LUT로 구현되므로 하나로 표현하였다. 기존의 방식에 비해 LUT의 개수가 줄어들어서 면적을 많이 줄일 수 있었다. 특히 로그 연산을 위한 LUT의 경우에는 입력 비트 수가 길어서 큰 LUT를 사용해야 하므로 로그 연산을 위한 LUT를 사용하지 않는 것이 큰 장점이다. 실행시간에서도 기존의 회로는 여러 단계를 거치는 반면에, 제안하는 회로는 한 번의 LUT 접근으로 계산이 가능하므로 실행 시간을 줄일 수 있다.

Table. 2 Area comparison (μm^2)

	Conventional1	Conventional2	Proposed
EXP	1658	4802	-
LOG	146436	146432	
Total	149839	152932	

V. 결론

본 논문에서는 CNN에 이용되는 SoftMax 층의 연산을 구현할 때 회로 복잡도를 줄이는 방법을 제안하였다. 기존에 사용되는 여러 개의 큰 LUT를 줄이기 위해, 입력 값 중에서 일부 큰 값들만을 선택한 뒤, 이들로부터 출력값을 직접 계산하는 LUT를 사용하였다. 이를 통해 계산의 정확도는 조금 낮아졌으나, 회로의 복잡도는 크게 낮아졌다. 이러한 SoftMax의 연산 회로는 다양한 CNN에서 널리 이용될 수 있을 것이다.

ACKNOWLEDGEMENT

This work was also supported by the 2020 Professor

Education and Research Promotion Program of KOREATECH, also supported by IDEC (EDA Tool), and also supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-004).

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems*, Lake Tahoe: NV, pp. 1097-1105, 2012.
- [2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of International Conference on Learning Representations*, San Diego: CA, pp. 1-14, 2015.
- [3] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Sog, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of Computer Vision and Pattern Recognition*, Honolulu: HW, pp. 1-21, 2017.
- [4] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of International Conference on Machine Learning*, Long Beach: CA, pp. 6105-6114, 2019.
- [5] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in *Proceedings of Computer Vision and Pattern Recognition*, online, pp. 10781-10790, 2020.
- [6] Y. J. Kim and E. G. Kim, "Image based Fire Detection using Convolutional Neural Network," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 20, no. 9, pp. 1649-1656, Sep. 2016.
- [7] B. Yuan, "Efficient hardware architecture of softmax layer in deep neural network," in *Proceedings of IEEE International System-on-Chip Conference (SOCC)*, Seattle: WA, pp. 323-326, Sep. 2016.
- [8] J. Park and D. Jeon, "Designing neuromorphic processor with on-chip learning," *IDEC Journal of Integrated Circuits and Systems*, vol. 6, no. 2, pp. 1-6, Apr. 2020.