임베디드 보드에서의 인공신경망 압축을 이용한 CNN 모델의 가속 및 성능 검증

문현철, 이호영, 김재곤
한국항공대학교
hcmoon@kau.kr, leehy9307@gmail.com, jgkim@kau.ac.kr

Acceleration of CNN Model Using Neural Network Compression and its Performance Evaluation on Embedded Boards

Hyeon-Cheol Moon, Ho-Young Lee, and Jae-Gon Kim
Korea Aerospace University

요 약

최근 CNN 등 인공신경망은 최근 이미지 분류, 객체 인식, 자연어 처리 등 다양한 분야에서 뛰어난 성능을 보이고 있다. 그러나, 대부분의 분야에서 보다 더 높은 성능을 얻기 위해 사용한 인공신경망 모델들은 파라미터 수 및 연산량 등이 방대하여, 모바일 및 IoT 디바이스 같은 연산량이나 메모리가 제한된 환경에서 추론하기에는 제한적이다. 따라서 연산량 및 모델 파라미터 수를 압축하기 위한 딥러닝 경량화 알고리즘이 연구되고 있다. 본 논문에서는 임베디트 보드에서의 압축된 CNN 모델의 성능을 검증한다. 인공지능 지원 맞춤형 칩인 QCS605 를 내장한 임베디드 보드에서 카메라로 입력한 영상에 대해서 원 CNN 모델과 압축된 CNN 모델의 분류 성능과 동작속도 비교 분석한다. 본 논문의 실험에서는 CNN 모델로 MobileNetV2, VGG16 을 사용했으며, 주어진 모델에서 가지치기(pruning) 기법, 양자화, 행렬 분해 등의 인공신경망 압축 기술을 적용하였을 때 원래의 모델 대비 추론 시간 및 분류의 정확도 성능을 분석하고 인공신경망 압축 기술의 유용성을 확인하였다.

1. 서론

최근 CNN 등 인공신경망은 괄목한 성능의 향상과 함께 영상 분류, 객체 인식, 미디어 압축, 데이터 분석, 자연어 처리 등 다양한 분야에 적용되고 있다. 인공신경망을 적용한 대부분의 응용분야에서는 상대적으로 연산량이 방대한 훈련된 인공신경망 모델을 연산능력과 메모리가 제한된 장치에 구현해야 한다. 따라서 학습된 모델의 연산량 및 모델 파라미터 수를 압축하기 위한 딥러닝 경량화 알고리즘이 연구되고 있다. 대표적인 경량 딥러닝 알고리즘으로는 가중치와 노드의 연결을 끊는 가지치기 기법과 각각의 가중치의 비트 값을 줄이는 양자화 기법, 그리고 한 층에서 가지고 있는 가중치 행렬을 분해하여 가중치 파라미터 수를 줄이는 행렬 분해기법 등이 있다. 본 논문에서는 최근 딥러닝 지원 맞춤형 칩인 QCS605 를 내장한 임베디드 보드에서 카메라로 입력한 영상에 대한 원 CNN 모델과 경량 딥러닝 알고리즘을 적용한 압축모델의 이미지 분류에 대한 추론(inference) 속도 및 분류 성능을 비교하였다.

2. 딥러닝 경량화 알고리즘

2.1 가지치기(Pruning)

가지치기 기법은 기존 신경망 모델에서 가지고 있는 가중치 중에서 비교적 작은 값들에 대한 가중치들을 불필요하다고 간주하여 해당 가중치 값을 0 으로 만드는 기술이다 [1]. 본 논문에서는 각 층에서 목표로 하는 희소도(sparsity)에 맞는 임계 값을 구하고 그 해당 값보다 낮은 값을 갖는 가중치들을 모두 0 으로 한다.

2.2 행렬 분해(Matrix Decomposition)

행렬 분해 기법은 가중치 행렬을 2 개 혹은 그 이상의 행렬로 분해하여 가중치 파라미터 수를 줄이는 기법이다. 예를 들어, 완전연결층(Fully-Connected layer)은 2-Dn 형태의 가중치 행렬이기 때문에 이를 행렬 2 개로 분해하는 낮은 순위 근사치 방법(Low-Rank Approximation)방법[2]을 이용할 수 있다. 그러나, 일반적인 컨볼루션 층에서는 일반적으로 N-D(여기서 N 은 N>=3 를 만족) 이상의 가중치 행렬이므로, 이를 N 개의 1-D 행렬로 분해하는 CP(CANDECOMP/PARAFAC) 분해 방법 등을 적용하여 Pointwise 와 Depthwise 컨볼루션 층으로 분해할 수 있다 [3].

3. QCS 605 임베디드 보드

Figure 1. QCS 605 칩(좌), 임베디드 보드 구성(우)

본 연구에서는 학습된 인공신경망 모델을 QCS605 SoC 를 이용한 임베디드 보드에서 포팅하고 성능 검증 실험을 진행하였다. 그림 1 은 해당 QCS 605 칩과 칩을 내장한 임베디드 보드 예시이다. 표 1 은 QCS605 의 연산능력을 보여주는 CPU 및 GPU 등의 규격을 나타낸 것이다[4]. 표 2 는 QCS605 를 이용한 임베디드 보드 QCS605 EV-R.2 Board 의 환경을 나타내었다.

Table 1. QCS 칩의 규격

AP	Specifications
CPU	Qualcomm® Kryo™ 300 CPU, Octa-core CPU / Up to 2.5 GHz / 64-bit Architecture
GPU	Qualcomm® Adreno™ 615 GPU PI Support: OpenGL® ES 3.2, Vulkan® 1.1, OpenCL
Memory	eMCP / LPDDR4x 4GB, 1866MHz

Table 2. QCS605 EV-R.2 보드의 규격

AP	Specifications
OS	Android 9.x / Kernel 4.x
Memory	eMMC: 64GB
Support I/O	2 MIPI CSI, 2 MIPI DSI / USB3.1 and DP / Audio input / output / SD Card I/F 1 port
WIFI /BT	802.11 a/b/g/n/ac, Tri-band support 802.11 ad / BT 5.0

본 논문의 실험을 위해 보드 내의 Android OS 환경에서의 Tensorflowlite 프레임워크를 이용하여 영상 분류 실험을 하였다. 실험은 보드 내의 카메라에 입력되는 영상을 인공신경망 모델인 tflite 모델로 추론하여 결과창에 상위 3 개의 분류에 대한 확률을 표시하도록 하였다.

4. 실험결과

본 논문에서의 실험은 VGG-16 과 MobileNetv2 의 모델과 해당 모델에 경량 딥러닝 알고리즘을 적용한 모델에 대하여 ImageNet 테스트 데이터로 측정하는 분류 성능 (Top-5 정확도)과 실제 보드에서의 카메라로 입력 받는 영상에 대한 모델 추론 속도를 비교하였다.

표 3 은 VGG-16 에서의 경량화 알고리즘 중 하나인 행렬 분해기법을 적용한 실험결과를 나타낸 것이다. 본 논문에서의 행렬 분해 기법은 완전연결층에서는 2 개의 행렬로 분해하는 낮은 순위 근사 기법을 적용하였고, 컨볼루션 층에는 N 개의 행렬로 분해하는 CP 분해기법을 적용하였다. 또한, 별도의 재학습(re-training)을 적용하지 않았다. 각 시험을 위해 중간 순위(Rank) 값을 조절하면서 압축율을 조정하였고, 해당 실험의 Test 는 약 5 배(Test 1/2)와 10 배(Test 3/4) 압축을 목표로 분해 기법을 적용하였다. ImageNet 데이터의 실험결과는 모든 시험에서 3%미만의 성능 감소를 보였으며, 추론 속도 측면에서는 원본 모델 대비 약 2 배 향상됨을 보여주었다.

표 4 는 MobileNetV2 에서의 가지치기 기법을 적용한 실험결과이다. 표 4 에서의 희소도는 전체 가중치 개수 대비 0 을 가지는 가중치의 비율이다. 가지치기 기법에서는 희소도가 높아질수록 그만큼 압축율이 높아지고, 그에 따른 성능 감소가 있다. 해당 실험에서는 모든 시험에서 Top-5 정확도 1~4%의 감소를 보여주었고, 실제 보드에서의 추론 속도는 거의 동일함을 보였다. 추론 속도가 거의 동일한 이유는 해당 인공신경망 모델 포맷의 한계로 가지치기 기법이 단순히 가중치를 0 으로만 만들었기 때문에 실질적인 연산량 감소의 효과를 기대할 수 없었다.

Table 3. 행렬분해기법을 적용한 실험결과(VGG-16)

Model	Model_size (MB)	Top-5 Accuracy (%)	Inference Time (ms)
VGG-16 original	527	89.95	2,040
VGG-16 Test1	121	89.43	1,020
VGG-16 Test2	110	88.25	1,000
VGG-16 Test3	66	88.29	970
VGG-16 Test4	50	86.75	900

Table 4. 가지치기 기법을 적용한 실험결과(MobileNetV2)

Model	Sparsity (%)	Top-5 Accuracy (%)	Inference Time (ms)
MobileNetV2 original	100	90.04	77
MobileNetV2 Test1	35	87.65	75
MobileNetV2 Test2	30	87.83	78
MobileNetV2 Test3	20	89.80	77
MobileNetV2 Test4	15	90.08	77

5. 결론

본 논문에서는 VGG-16 과 MobileNetV2 의 모델에 가지치기 및 행렬 분해 기법을 적용하여 실제 분류 성능과 보드에서의 추론속도를 비교하였다. 행렬 분해 기법은 파라미터 수를 줄여줄 뿐만 아니라 실제 보드에서의 추론 속도 향상의 효과를 보여주었으며, 실제 분류성능도 1.5%미만의 Top-5 정확도 감소를 보여주었다. 반면에 가지치기 기법에서는 행렬 분해 기법보다 더 큰 분류 성능 감소와 함께 보드에서의 추론 속도도 거의 동일하였다.

추후에 가지치기 기법에 대해서도 실질적인 추론 속도 감소 및 실질적인 모델 크기를 줄일 수 있도록 개선할 예정이다. 또한 더 다양한 모델과 딥러닝 경량 알고리즘을 개선하여 연산환경이 제한된 모바일이나 IoT 기기에서도 인공신경망 모델을 이용한 응용분야에 도움이 될 것을 기대한다.

Acknowledgement

이 논문은 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2017R1D1A1B03030331).

References

[1] S. Han, et al, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” ICLR, Oct. 2015.

[2] H. Moon, S. Chun, and J. Kim, “KAU/Insignal Response to the NNR CE-2 on Neural Network Compression: Low-Rank Approximation (Test 4),” ISO/IEC JTC1/SC29/WG11, m48885, July 2019.

[3] H. Moon, J. Kim, S. Kim, S. Jang, and B. Choi, “CE-2 on Neural Network Compression Related: CP-Decomposition of Convolutional Layer,” ISO/IEC JTC1/SC29/WG11, m48887, July 2019.

[4] [Available at Online] https://www.qualcomm.com/products/qcs605.