ETRI

2019
Electronics and
Telecommunications
Trends

경량 딥러닝 기술 동향

Recent R&D Trends for Lightweight Deep Learning

이용주 (Y.J. Lee, yongju@etri.re.kr) 스마트데이터연구그룹 책임연구원/PL
문용혁 (Y.H. Moon, yhmoon@etri.re.kr) 스마트데이터연구그룹 선임연구원
박준용 (J.Y. Park, junyong.park@etri.re.kr) 스마트데이터연구그룹 연구원
민옥기 (O.G. Min, ogmin@etri.re.kr) 스마트데이터연구그룹 책임연구원/그룹장

Considerable accuracy improvements in deep learning have recently been achieved in many applications that require large amounts of computation and expensive memory. However, recent advanced techniques for compacting and accelerating the deep learning model have been developed for deployment in lightweight devices with constrained resources. Lightweight deep learning techniques can be categorized into two schemes: lightweight deep learning algorithms (model simplification and efficient convolutional filters) in nature and transferring models into compact/small ones (model compression and knowledge distillation). In this report, we briefly summarize various lightweight deep learning techniques and possible research directions.

* DOI: 10.22648/ETRI.2019.J.340205

* This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) [No.2018-0-00278, Development of Big Data Edge Analytics SW Technology for Load Balancing and Active Timely Response].

Ⅰ. 서론

최근 들어, 이미지, 소리, 텍스트 형태로 이루어진 무한한 양의 빅데이터를 이해하기 위하여 강력한 GPU(Graphics Processing Unit) 기반의 컴퓨팅 자원을 바탕으로 학습을 통해 다양한 딥러닝 모델이 만들어지지만, 경량 디바이스, 모바일 디바이스, 산업용 게이트웨이, IoT 센서와 같은 디바이스에서 직접 학습과 추론이 가능할 정도의 수준은 미미하여, 실제 지능형 디바이스로 변화하기에는 역부족이다. 이러한 추세로 기존의 학습된 모델의 정확도를 유지하면서 보다 크기가 작고, 연산을 간소화하는 연구인 경량 딥러닝 연구가 활발히 진행되고 있다. 경량 딥러닝 연구는 기존 클라우드 기반의 학습된 모델을 경량 장치에 내장하기 위한 필수 기술이며, 이를 통해 지연시간 감소, 민감한 개인 정보 보호, 네트워크 트래픽 감소 같은 다양한 이점을 갖게 된다.

경량 딥러닝 기술은 알고리즘 자체를 적은 연산과 효율적인 구조로 설계하여, 기존 모델 대비 효율을 극대화하기 위한 경량 딥러닝 알고리즘 연구와 만들어진 모델의 파라미터들을 줄이는 모델 압축(Model Compression) 등의 기법이 적용된 알고리즘 경량화 기술로 나눌 수 있다.

경량 딥러닝 알고리즘은 가장 일반화된 합성곱 신경망(CNN: Convolutional Neural Network)을 통해 다양한 연구가 진행 중이다. CNN 계열의 모델에서는 주로 학습 시 가장 큰 연산량을 요구하는 합성곱 연산을 줄이기 위한 효율적인 합성곱 필터 기술이 일반화되고 있다. 다양한 신규 계층 구조를 설계하여 신경망 구조를 제공함으로써 우수한 추론 성능을 보이는 연구도 소개되고 있다. 이는 기본 단일 층별 연산에 그치지 않고 연산량과 파라미터 수를 줄이기 위한 잔여 블록(Residual Block) 또는 병목 블록(Bottleneck Block)과 같은 형태를 반복적으로 쌓아 신경망을 구성하는 방법이다. 마지막으로 기존 신경망의 모델 구조를 인간에게 의존적으로 수행하지 않고 모델 구조를 자동 탐색함으로써 모델을 자동화하거나 연산량 대비 모델 압축 비율을 조정하는 등 다양한 자동 탐색 기술이 존재한다. 이는 모바일 딥러닝과 같은 다양한 기기의 성능 대비 추론 속도가 중요한 응용을 위해 정확도, 지연시간, 에너지 소모량들을 사용하여 강화 학습(Reinforcement Learning)을 활용하여 경량 모델을 탐색하는 기술이다.

알고리즘 경량화는 경량 딥러닝 알고리즘과 달리, 모델이 표현하는 다양한 파라미터의 크기를 줄이는 데 주목적을 가지고 있다. 파라미터가 가지는 표현력을 가능한 한 유지하면서 불필요한 가중치를 최대한 없애기 위한 방법들이다. 일반적인 딥러닝 모델은 과파라미터(Over-parameterization)화 되어 있기 때문에 모델이 가지는 가중치의 실제값이 아주 작을 경우, 모델의 정확도에 큰 영향을 미치지 못하므로(이를 모델이 작은 가중치에 대한 내성을 가진다고 표현함), 이 값을 모두 0으로 설정하여 마치 가지치기(Pruning)를 수행하는 것과 같은 효과를 내는 가중치 가지치기(Weight Pruning)가 대표적이다. 다음으로, 일반적인 모델의 가중치는 부동 소수점값을 가지지만, 이를 특정 비트 수로 줄이는 양자화(Quantization)를 통해 기존 딥러닝의 표현력을 유지하면서 실제 모델의 저장 크기는 줄이는 방법이 있다. 마지막으로, 0과 1로 표현하여 표현력을 많이 줄이지만, 정확도의 손실은 어느 정도 유지하면서 모델 저장 크기를 확연히 줄이는 이진화(Binarization) 기법 등이 있다.

경량 딥러닝 기술은 크게 딥러닝 모델의 구조적 한계를 극복하고자 하는 경량 딥러닝 알고리즘과 기존 모델의 효율적인 사용을 위한 알고리즘 경량화의 두 축으로 연구가 진행되고 있다.

〈표 1〉 경량 딥러닝(Lightweight Deep Learning) 연구 동향

	접근방법	연구 방향
경량 알고리즘 연구	모델 구조 변경	잔여 블록, 병목 구조, 밀집 블록 등 다양한 신규 계층 구조를 이용하여 파라미터 축소 및 모델 성능을 개선하는 연구(ResNet, DenseNet, SqueezeNet)
	합성곱 필터 변경	합성곱 신경망의 가장 큰 계산량을 요구하는 합성곱 필터의 연산을 효율적으로 줄이는 연구(MobileNet, ShuffleNet)
	자동 모델 탐색	특정 요소(지연시간, 에너지 소모 등)가 주어진 경우, 강화 학습을 통해 최적 모델을 자동 탐색하는 연구(NetAdapt, MNasNet)
알고리즘 경량화 연구	모델 압축	가중치 가지치기, 양자화/이진화, 가중치 공유 기법을 통해 파라미터의 불필요한 표현력을 줄이는 연구(Deep Compression, XNOR-Net)
	지식 증류	학습된 기본 모델을 통해 새로운 모델의 생성 시 파라미터값을 활용하여 학습시간을 줄이는 연구(Knowledge Distillation, Transfer Learning)
	하드웨어 가속화	모바일 기기를 중심으로 뉴럴 프로세싱 유닛(NPU)을 통해 추론 속도를 향상시키는 연구
	모델 압축 자동 탐색	알고리즘 경량화 연구 중 일반적인 모델 압축 기법을 적용한 강화 학습 기반의 최적 모델 자동 탐색 연구(PocketFlow, AMC)

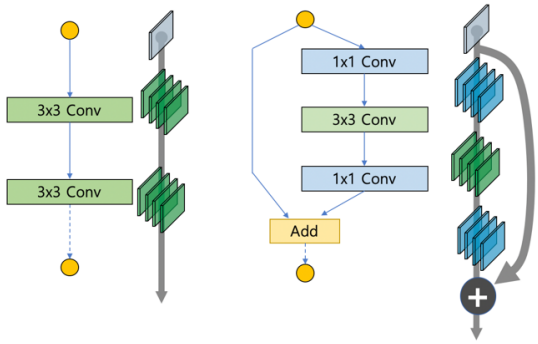
II. 경량 딥러닝 알고리즘

합성곱 신경망은 처음으로 여러 개의 합성곱 층과 활성화 함수를 연속적으로 이어 붙인 알렉스넷(AlexNet)을 시작으로 합성곱 층 이후에 다운 샘플링을 통해 통과하는 격자의 크기를 줄여 연산량과 변수가 많아 학습되지 않는 문제점을 해결하고자 하였다. 이후 ZF-Net, VGGNet을 거치면서 점차 필터의 크기가 줄어들어서 1×1 필터를 주로 사용하였다. 필터의 축소 이외에 단일 필터를 사용하는 구조에서 벗어나 서로 다른 필터를 병렬로 연결하는 인셉션(Inception) 모듈을 통해 다양한 형태로 발전하였다. Inception 모델 형태는 v1, v2, v3를 거치면서 다양한 형태로 연구가 진행되었다. 이러한 연구가 레즈넷(ResNet)과 같이 두 개의 연속적인 합성곱 층에 단위행렬의 추가를 위한 지름길(shortcut)을 더해 줌으로써 가중치들이 더 쉽게 최적화될 수 있는 잔여 블록(Residual Block) 형태로 개선되었으며, 점차 병목 구조(Bottleneck Architecture), 밀집 블록(Dense Block) 형태로 발전되고 있다.

1. 모델 구조 변경 기술

가. 레즈넷

깊은 신경망의 문제는 층의 수가 늘어나면서 점차 정확도가 저하되는 문제가 발생하는데, 레즈넷(ResNet)[1], [2]은 (그림 1)의 왼쪽과 같이 계층(Weight layer)이 계속 쌓이는 경우, 최적의 H(x)를 찾는 문제에서 이를 (그림 1)의 오른쪽과 같이 문제의 정의를 바꾸어 출력과 입력의 차이(H(x)-x=F(x))를 목표로 하면, 출력은 F(x)+x가 된다. 결국, 지름길(x)을 통해 파라미터 없이 바로 연결



(그림 1) 기존 평형망과 잔여 신경망의 비교

되는 구조로 바꾸고, 연산량 관점에서 덧셈이 추가되는 형태로 문제를 단순화할 수 있게 된다. 이러한 나머지, 즉 잔여(Residual, F(x))를 학습하는 형태로 발전하게 된다. 이를 통해 깊은 신경망에도 쉽게 최적화가 가능하며, 늘어난 깊이로 인한 정확도 개선 효과도 보게 된다.

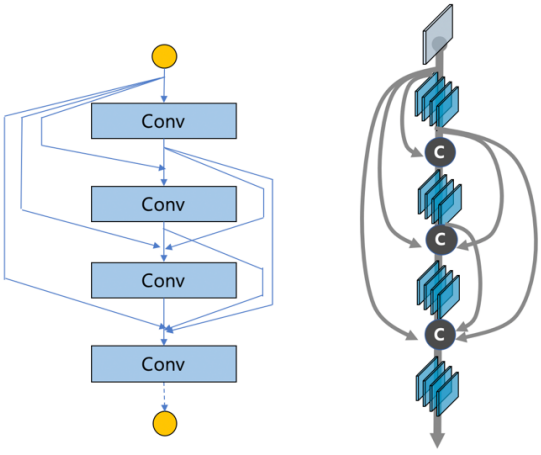
나. 덴스넷

기존 신경망 모델 구조의 여러 장점을 모아 덴스넷(DenseNet)[3]이 고안되었다. 기존 피처맵(feature map)을 더해 주는 게 아닌 쌓아가는 과정을 거치며 모델의 성능을 높이고자 하였다[(그림 2) 참조].

아울러, 이미지 판별 문제(Image Classification)의 경우, 맨 마지막 층의 하이 레벨 피처에서만 사용하던 것이, 덴스넷에서는 이전의 모든 층에서의 정보를 취득하는 형태가 가능하다. 이를 통해, 기존의 다른 네트워크보다 굉장히 좁게 설계 가능해지고, 파라미터 수를 줄일 수 있게 되었다.

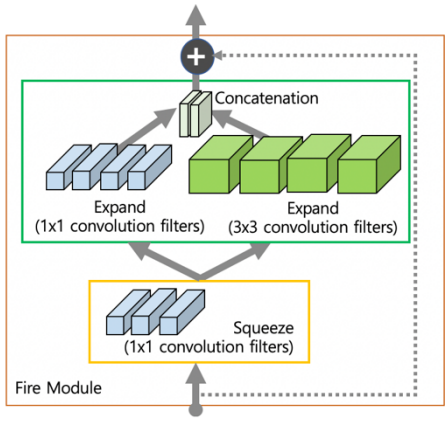
다. 스퀴즈넷

스퀴즈넷(SqueezeNet)[4]은 기본적으로 사용하는 합성곱 필터인 3×3 필터를 1×1 필터로 대체함



(그림 2) 덴스넷(DenseNet)의 밀집 신경망 구조

[출처] Reproduced from Zhuang Liu, https://github.com/liuzhuang13/DenseNet



(그림 3) 스퀴즈넷(SqueezeNet)의 파이어 모듈(Fire Module)

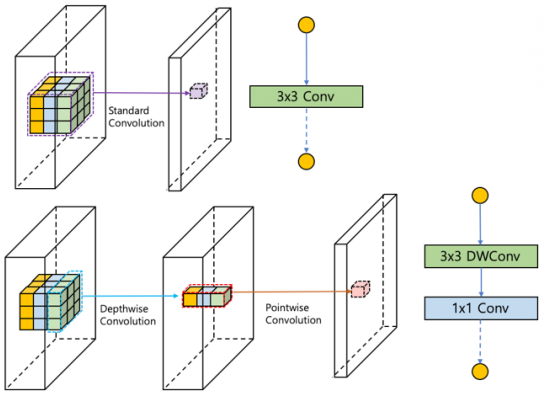
으로써 9배 적은 파라미터를 가지며, 1×1 합성곱을 이용하여 채널 수를 줄였다가 다시 늘리는 파이어 모듈(Fire Module) 기법을 제안하였다. 또한, 늦은 다운 샘플링 전략을 통해 한번에 필터가 볼 수 있는 영역을 좁히면서 해당 이미지의 정보를 압축시키는 효과를 볼 수 있게 된다[(그림 3) 참조].

2. 효율적인 합성곱 필터 기술

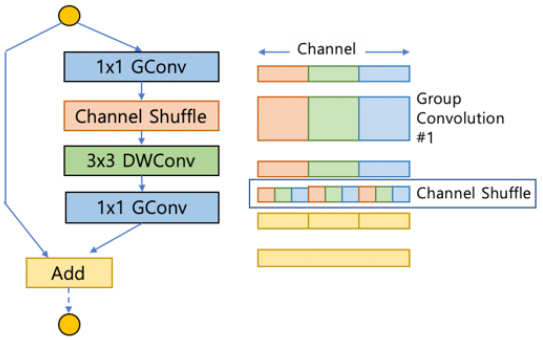
모델 구조를 변경하는 다양한 경량 딥러닝 기법은 점차 채널을 분리하여 학습시키면서 연산량과 변수의 개수를 줄일 수 있는 연구로 확장되었다. 일반적인 합성곱은 채널 방향으로 모두 연산을 수행하여 하나의 특징을 추출하는 데 반해, 채널별(Depthwise)로 합성곱을 수행하고, 다시 점별(Pointwise)로 연산을 나누어 전체 파라미터를 줄이는 것과 같이 다양한 합성곱 필터를 설계하기 시작하였다. 이후, 점별 그룹 형태로 섞는 셔플 방법이 연구 진행 중이다.

가. 모바일넷

모바일넷(MobileNet)[5], [6]에서는 기존의 합성곱 필터를 채널(Channel) 단위로 먼저 합성



(그림 4) 모바일넷(MobileNet)의 합성곱 분해 구조



(그림 5) 셔플넷(ShuffleNet)의 채널 셔플 구조

[출처] Reprinted with Permission from https://arxiv.org/abs/1707.01083, 2017.

곱(Depthwise Convolution)을 하고, 그 결과를 하나의 픽셀(Point)에 대하여 진행하는 합성곱(Pointwise Convolution)으로 나눔으로써 한 예로, 필터의 가로, 세로 길이를 3이라고 할 때, 약 8~9배의 이득이 있게 하였다[(그림 4) 참조].

나. 셔플넷

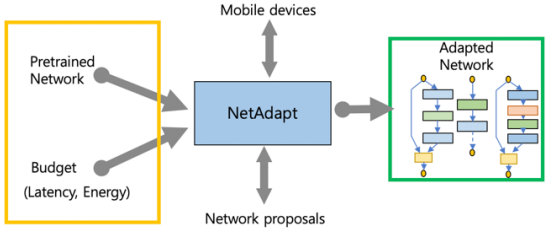
액셉션(Xception)이나 모바일넷(MobileNet)에 제안된 채널별(Depthwise) 개별 합성곱은 표현 성능은 그리 높지 않지만, 연산량을 대폭 줄일 수 있기 때문에 현재까지도 많이 사용되고 있으며, 셔플넷(ShuffleNet)[7], [8]에서는 점별 합성곱(Pointwise convolution) 시 특정 영역의 채널에 대해서만 연산을 취하는 형태로 설계하면 연산량을 매우 줄일 수 있을 것으로 생각하였다. 입력에서만의 정보 흐름만을 취하는 대신, 입력의 각 그룹이 잘 섞일 수 있도록 개선한 것이 핵심이다[(그림 5) 참조].

3. 경량 모델 자동 탐색 기술

최근 들어, 강화 학습 기법이 적용된 다양한 응용이 활발히 연구되고 있으며, 모델 구조와 합성곱 필터를 설계하고 구현하는데, 강화 학습을 통한 자동 탐색 기법들이 소개되고 있다. 이는 기존의 신경망의 최적화는 MACs(Multiplier-Accumulators) 또는 FLOPs(Floating Operations Per Seconds)에 의존하였으나, 실용적인 방식인 Latency 또는 Energy Consumption 문제로 기준이 바뀌고 있다. 그에 따라, 추론에 최적화된 신경망을 자동 생성하거나 연산량 대비 모델의 압축비를 조정하는 데 사용되고 있다. 또한 신경망을 생성, 훈련, 배포하는 과정을 크게 단축시키는 역할을 하고 있다.

가. 넷어탭트

넷어탭트(NetAdapt)[9]는 주어진 Budget을 만족하는 최적의 결과를 얻고자 하며, 이를 만족하기



(그림 6) 넷어탭트(NetAdapt)의 신경망 탐색 흐름

[출처] Reprinted with Permission from https://arxiv.org/abs/1804.03230, 2018.

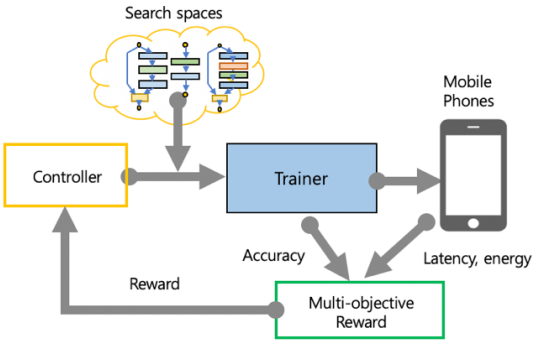
위해 주어진 Budget을 여러 개로 나누어 한번에 일정 만큼씩 만족하는 조건을 점차 찾는 방식이다[(그림 6) 참조].

나. 엠나스넷

신경망 아키텍처 탐색을 위해 모바일 환경에서 탐색 알고리즘의 메인 보상 함수에 속도 정보를 명시적으로 포함하여 정확도와 속도 간의 균형을 이루는 모델을 탐색하는 방식으로 기존 나스넷(NasNet)보다 2.4배 빠르게 실행되는 모델을 찾을 수 있다. 엠나스넷(MNasNet)[10]은 기존 모델 아키텍처 학습과 샘플링을 위한 순환 신경망(RNN: Recurrent Neural Network)을 기반으로 한 컨트롤러, 모델을 만들고 훈련시켜 정확도를 얻는 트레이너, 실제 모바일폰에서 추론 엔진을 수행하여 얻은 추론 지연 시간을 통해 다(多) 목표 최적화 문제를 만들고 보상 함수가 포함된 강화 학습 알고리즘을 통해 파레토(Pareto) 최적 솔루션을 찾는 방식이다[(그림 7) 참조].

Ⅲ. 알고리즘 경량화

알고리즘 경량화 연구는 효율적인 네트워크 구조를 설계하거나 합성곱 연산의 다양한 변이, 모델 자동 탐색과 같은 경량 알고리즘의 연구와 달리, 기존 알고리즘의 불필요한 파라미터를 제거하거나, 파라미터의 공통된 값을 가지고 공유하거나, 파라미터의 표현력을 잃지 않으면서 기존 모델의 크기를 줄이는 연구 분야이다. 주로 모델 압축(Model Compression) 및 지식 증류(Knowledge Distillation) 연구와 가속화(Acceleration) 형태로 진행되고 있다.



(그림 7) 엠나스넷(MNasNet)의 신경망 탐색 흐름

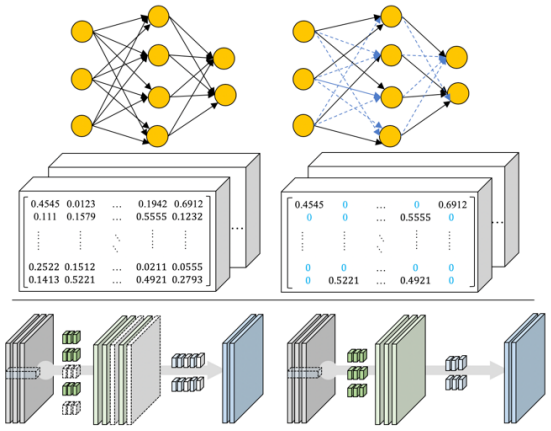
[출처] Reprinted with Permission from https://arxiv.org/abs/1807.11626, 2018.

1. 모델 압축 기술

가. 가중치 가지치기

기존 신경망이 가지고 있는 가중치(Weights) 중 실제 추론을 위해 필요한 값은 비교적 작은 값들에 대한 내성을 가지므로, 작은 가중치값을 모두 0으로 하여 네트워크의 모델 크기를 줄이는 기술이다.

이후의 연구는 가중치 가지치기(Weight Pruning) 후에 재훈련 과정을 통해 정확도를 높일 수 있는 방식으로 신경망을 세밀하게 조율하는 방식으로 진행되고 있다. 또한, 일반적인 가중치를 통한 접근 방법 이외에 채널을 선별하여 중복(불필요한) 채널에 대한 가지치기를 통해 모델을 압축하는 연구도 진행 중이다[(그림 8) 참조], [11].



(그림 8) 가중치/채널 가지치기(Weight/Channel Pruning)의 예

[출처] Reprinted with Permission from https://arxiv.org/abs/1510.00149, 2015 and, https://arxiv.org/abs/1707.06168, 2017.

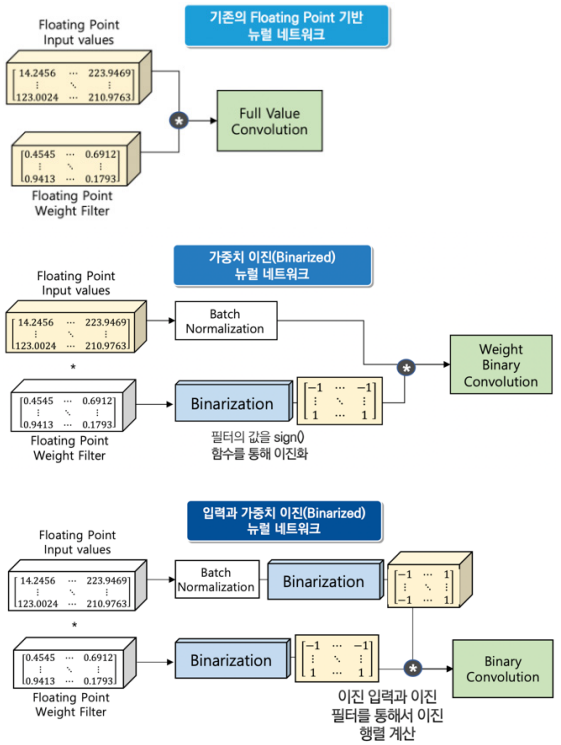
나. 양자화 및 이진화

양자화와 이진화는 기존의 신경망의 부동 소수점 수를 줄이는 데 그 목적이 있으며, 양자화(Quantization)의 경우 특정 비트 수만큼으로 줄여서 계산하는 방식이다[(그림 9) 참조].

가령, 32비트 소수점을 8비트로 줄여서 연산을 수행한다. 이진화(Binarization)는 신경망이 가지고 있던 가중치(Weights)와 층 사이의 입력을 부호에 따라서 −1 혹은 +1의 이진(Binary) 형태의 값으로 변환하여, 기존의 Floating Point를 사용하는 신경망들에 비해 용량과 연산량을 대폭 압축시키는 기술이다[12].

다. 가중치 공유

가중치 공유(Weight Sharing) 기법은 낮은 정밀도에 대한 높은 내성을 가진 신경망의 특징을 활용해 가중치를 근사하는 방법이다. 기존 가중치값들은 근사한 값(코드북)을 통해 가중치들을 공유하는데, 코드북과 그 값에 대한 인덱스만을 저장하는 구조이므로, 실제 저장 공간을 절약할 수 있다. 근사화하는 방식은 가중치들의 유사도에 기반하는데, 주로 K-Means 또는 Gaussian Mixture Model[13]을 활용한다.

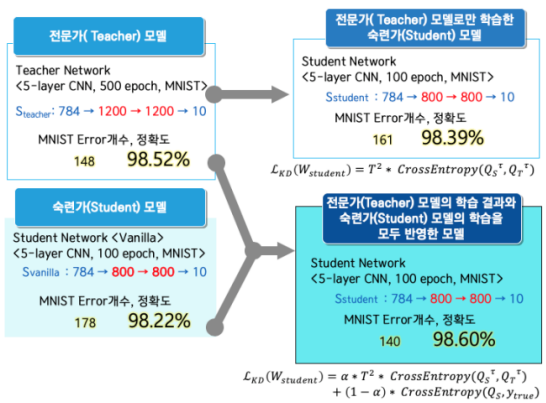


(그림 9) 이진화(Binarization)를 통한 합성곱의 예

2. 지식 증류 기술

지식 증류(Knowledge Distillation)[14] 기술은 앙상블(Ensemble) 기법을 통해 학습된 다수의 큰 네트워크들로부터 작은 하나의 네트워크에 지식을 전달할 수 있는 방법론 중의 하나이다. 다수의 큰 네트워크들인 전문가(Experts, Teacher) 모델에서 출력은 일반적으로 특정 레이블에 대한 하나의 확률값 만을 나타내지만, 이를 확률값들의 분포 형태로 변형하여, 숙련가(Specialist, Student) 모델의 학습 시에 모델의 Loss와 전문가 모델의 Loss를 동시에 반영하는 형태로 숙련가 모델을 학습에 활용한다[(그림 10) 참조].

일반적인 지식 증류 기술의 연구는 모델 압축 기



(그림 10) 전문가(Teachers) 모델과 숙련가(Student) 모델의 학습 결과 예

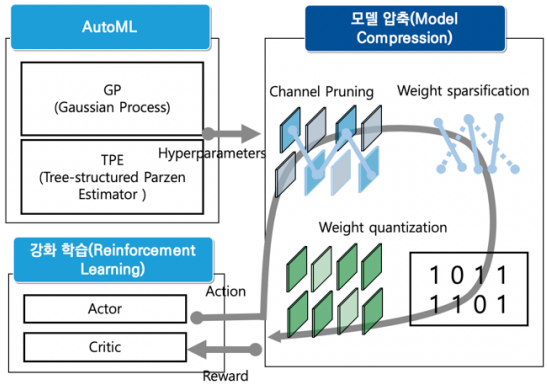
술과 같이 신경망을 간소화하는 방식으로 이루어지고 있지만, 반대로 현재 훈련된 네트워크보다 더 큰 네트워크로의 지식 전이(Knowledge Transfer)[15]를 하는 연구도 진행 중이다. 이러한 지식 전이 기법 중에서 더 깊거나 더 넓은 네트워크를 만들 때 정보를 완벽히 동일하게 전달하는 연산(Operation) 방법도 가능하다.

3. 하드웨어 가속화 기술

벡터/행렬 연산을 병렬 처리하기 위한 전용 하드웨어 TPU(Tensor Processing Unit), On-Device AI 응용 추론을 위한 전용 VPU(Visual Processing Unit) 프로세스 및 GPU Cluster 기반 가속기 등의 연구 개발이 주요 IT 기업에 의해 주도되고 있다. 최근에는 경량 디바이스에 사용 가능한 칩셋 또는 USB 스틱 형태로 임베디드 장치에서 추론 연산 가속화가 이루어지고 있다. 대표적으로 인텔에서는 모비디우스를 통한 뉴럴 컴퓨트 스틱, 엔비디아에서는 젝슨 TX2, 구글에서는 엣지 TPU가 개발 중이다. 모바일 환경에서의 신경망 처리를 위한 모바일 AP는 퀄컴의 스냅드래곤, 화웨이의 기린, 애플의 A12칩, 삼성의 액시노스가 대표적인 사례이다. 향후에는 신경망 전용 프로세서가 장착된 다양한 경량 디바이스에서 추론이 가능할 것으로 예상된다.

4. 모델 압축을 적용한 경량 모델 자동 탐색 기술

경량 딥러닝 알고리즘의 모델 설계를 자동화하기 위하여 합성곱 연산, 커널 크기, 필터 크기, 층(Layer)의 개수 등 다양한 탐색 공간을 통한 다양한 모델을 설계하는 것과 유사하게, 알고리즘 경량화도 모델 압축 기법의 네트워크 가지치기, 가중치 양자화 등의 탐색 공간을 통한 자동화 연구가 진행



(그림 11) 강화학습을 통해 모델 압축/가속화 기법들을 자동 탐색하는 예

중이다.

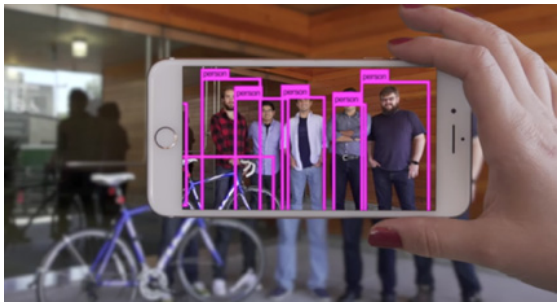
텐센트(Tencent)의 포켓플로(PocketFlow)[16]는 하이퍼 파라미터의 최적화를 통해 기존의 경량 알고리즘인 모바일넷(MobileNet)에 모델 압축 기법을 적용하고 있다.

또 다른 연구[17]는 정확도와 지연시간을 모두 고려한 강화학습 기반의 모델 압축 기법을 자동 탐색하는 기법도 소개되고 있다[(그림 11) 참조].

이러한 경량 알고리즘의 자동 탐색(NAS: Neural Architecture Search 또는 AutoML: Automated Machine Learning)에 그치지 않고 모델 압축 자동 탐색(AutoMC: Automated Model Compression) 형태로 진화하고 있는 추세이다.

Ⅳ. 경량 딥러닝 산업 동향

경량 딥러닝을 적용한 산업은 다양한 분야에서 현재 태동 단계에 있으며, 점차적으로 딥러닝 프레임워크도 경량 알고리즘을 위한 기법을 적용하고 있다. 하드웨어의 경우 다양한 엣지 디바이스 형태로 추론이 가능한 모바일 기기에서부터 산업 현장의 게이트웨이까지 적용 범위를 넓혀가고 있는 추세이다.



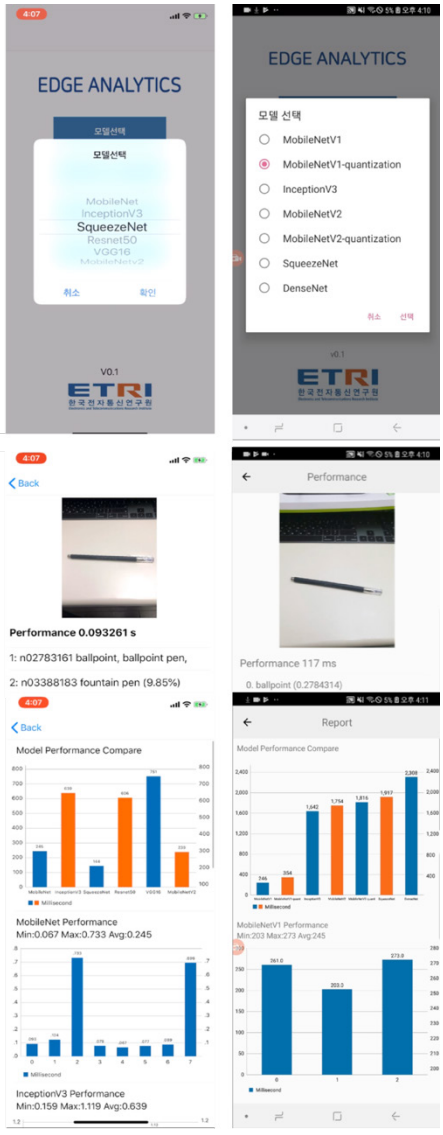
(그림 12) 스마트폰에서의 객체 인식 예

[출처] Reprinted with Permission from XNOR.AI, https://www.youtube.com/watch?v=ngHuVggHagg, 2017.

최근의 딥러닝 프레임워크 중 파이토치(PyTorch)와 텐서플로(Tensorflow)의 경우 양자화를 통한 경량화를 지원하고 있다. 주로 모바일 디바이스에 적용 가능한, 사전 훈련된 모델을 제공한다. 파이토치(PyTorch)의 경우, 모바일넷, 스퀴즈넷, 덴스넷 등이 대표적이며, 텐서플로의 경우 모바일넷, 엠나스넷, 스퀴즈넷 등을 제공한다.

해외 스타트업 중 XNOR.AI[18]의 경우, (그림 12)와 같이 이진화를 통한 객체 인식 기술을 적용하고 있으며, 국내의 경우 하이퍼커넥트[19]의 양자화를 적용한 이미지 세그멘테이션이 대표적이며, 한국전자통신연구원에서는 경량 디바이스에 기반한 다양한 엣지 분석 기술을 개발 중이다[(그림 13) 참조].

향후에 응용 가능한 사례는 모바일 기기 위주의 서비스 형태로 발전하고 있으며, 감정 분석 및 문장 번역 서비스, 이미지 분류 및 음악 태깅, 키보드 문자열 예측과 손글씨 분석과 같은 모바일 기기에서 사용자의 다양한 요구를 만족시키며, 즉시 대응 가능한 개인화 서비스가 대표적인 사례이다. 헬스 케어 분야에서는 카메라를 통한 안과 질환 검증, 피부암 진단과 같은 의료 데이터를 통한 온디바이스 AI가 가능해지고 있다. 자율주행 자동차 분야에서는 차량 카메라를 통한 눈동자 감김이나



(그림 13) 스마트폰(iOS/Android)에서의 경량 딥러닝 모델들의 이미지 판별 실험 예

[출처] ETRI(빅데이터 엣지 분석 기술)

고개 젖힘 감지, 내장 센서를 통한 운전 패턴 인식과 같은 다양한 서비스가 가능하다.

V. 결론

경량 딥러닝 기술은 폭발적으로 증가하는 다양

한 딥러닝 관련 기술 중 기존의 정확도(Accuracy)를 높이는 효율적인 신경망 구조를 만드는 데 그치지 않고, 실제 산업 현장에 적용하기 위한 다양한 형태의 간소화 및 경량화 기법들이 제시되고 있다. IoT 디바이스, 스마트폰 및 산업용 경량 장치에 탑재 가능한 모델의 형태는 경량 딥러닝 알고리즘을 통한 적은 파라미터를 가진 효율적인 구조에 대한 연구와 기존의 알고리즘의 불필요한 표현력을 줄이는 연구로 진행되고 있다. 향후에는 이러한 경량 딥러닝 기술을 응용한 보다 다양한 분야와 접목되어 실생활 곳곳에서 다양한 AI 기술이 상용화될 것이다.

용어해설

경량 딥러닝(Lightweight Deep Learning) 기존의 딥러닝을 통해 생성된 모델을 다양한 기법으로 줄여서(예. 크기, 에너지 소모 등) 정확도를 유지하면서 다양한 경량 디바이스에서 내장하여 추론을 가능하게 하는 기술

이진화(Binarization) 딥러닝에 사용되는 가중치의 부동 소수점 수를 0과 1로 표현하여, 그 크기를 줄이는 방법

양자화(Quantization) 0과 1로 표현하는 이진화와 달리, 특정 몇 비트(예. 8비트)로 구간화하여 그 크기를 줄이는 방법

지식 증류(Knowledge Distillation) 심층 신경망으로 학습된 모델들의 숨은 지식을 계산량이 적고 얕은 신경망으로 전달하는 방법

전이 학습(Transfer Learning) 기존의 학습된 모델과 비슷한 유형의 다른 모델로 학습된 결과를 옮겨서 부족한 데이터를 통한 학습이나 훈련 시간을 단축시키는 방법

약어 정리

AutoMC	Automated Model Compression
AutoML	Automated Machine Learning
CNN	Convolutional Neural Network
FLOP	Floating Operations Per Second
GPU	Graphics Processing Unit
MAC	Multiplier-Accumulator
NAS	Neural Architecture Search
RNN	Recurrent Neural Network
TPU	Tensor Processing Unit
VPU	Visual Processing Unit

참고문헌

[1] K. He et al., "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, Las Vegars, NV, USA, June 2016, pp. 770-778.

[2] K. He et al., "Identity Mappings in Deep Residual Networks," in *European Conference on Computer Vision*, Springer, 2016, pp. 630-645

[3] G. Huang et al., "Densely Connected Convolutional Networks," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Honolulu, HI, USA, July, 2017, pp. 2265-2269.

[4] F.N. Iandola et al., "SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and < 0.5MB model size," arXiv:1602.07360, 2016.

[5] A.G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861, 2017.

[6] M. Sandler et al., "MobileNet V2: Inverted Residuals and Linear Bottlenecks," arXiv:1801.04381, 2018.

[7] X. Zhang et al., "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," arXiv:1707.01083, 2017.

[8] M. Ningning et al., "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design," arXiv:1807.11164, 2018.

[9] T.J. Yang et al., "NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications," arXiv:1804.03230, 2018.

[10] M. Tan et al., MnasNet: Platform-Aware Neural Architecture Search for Mobile," arXiv:1807.11626, 2018.

[11] S. Han, H. Mao, and W.J. Dally, "Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding," arXiv:1510.00149, 2015.

[12] M. Rastegari et al., "XnorNet: ImageNet Classification Using Binary Convolutional Neural Networks," arXiv:1603.05279, 2016.

[13] K. Ullrich, E. Meeds, and M. Welling, "Soft Weight-Sharing for Neural Network Compression," arXiv:1702.04008, 2017.

[14] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," arXiv:1503.02531, 2015.

[15] T. Chen, I. Goodfellow, and J. Shlens, "Net2Net: Accelerating Learning via Knowledge Transfer," in *Int. Conf. Learning Representation (ICLR)*, May 2016.

[16] J. Wu, J. Hou and W. Liu, "PocketFlow : An Automated Framework for Compressing and Accelerating Deep Neural Networks,". in *Proc. Neural Inf. Process. Syst. (NIPS)*, Montreal, Canada, Dec. 2018.

[17] Y. He et al., :AMC: AutoML for Model Compression and Acceleration on Mobile Devices," in *Proc. Eur. Conf. Comput. Vision (ECCV)*, Munich, Germany, Sept. 2018, pp. 784 – 800.

[18] https://www.xnor.ai/

[19] https://hyperconnect.com/