슬라이싱과 Top-k 기반 피쳐 선택 기법을 활용한 동적 프루닝

최성헌○, 이하윤, 신동군
성균관대학교 전자전기컴퓨터공학과
csh9580@skku.edu, lhy920806@gmail.com, dongkun@skku.edu

Dynamic pruning technique utilizing both slicing and Top-k feature selection

Sunghern Choi○, Hayun Lee, Dongkun Shin
Department of Electrical and Computer Engineering, Sungkyunkwan University

요 약

하드웨어가 제한된 모바일 장치에서 딥 러닝 모델을 효율적으로 사용하기 위해서는 필요한 연산을 모바일 장치와 서버로 분할시켜야 한다. 필요한 연산을 분할 할 때 모바일 장치에서 일부 연산한 데이터를 서버로 전송시킨다. 서버로 데이터를 전송할 때 발생하는 지연시간이 크기 때문에 전송하는 데이터의 크기를 줄일 필요가 있다. 데이터의 크기를 줄이기 위한 JPEG과 같은 압축기법은 정확도의 손실이 크다. 따라서 본 논문에서는 정확도의 손실이 크지 않도록 압축을 하기 위한 프루닝 기법을 제안한다. 제안한 프루닝 기법을 사용해서 최대 50배를 압축했을 때의 정확도가 압축을 안 했을 경우와 비교했을 때 정확도 손실이 8% 이하인 것을 보여준다.

1. 서 론

최근 모바일 장치에서 딥 러닝 모델을 사용할 때 효율적으로 연산을 분할 하는 연구들이 많이 진행되었다[1, 2]. 연산을 분할 할 때는 모바일 장치에서 일부 연산을 처리하고 연산의 결과인 피쳐 맵(feature map)을 서버로 전송해서 남은 연산을 처리한다. 딥 러닝 모델의 연산을 분할 할 때 가장 중요한 점은 피쳐 맵을 전송할 때 발생하는 지연시간을 줄이는 것이다. 이를 위해 피쳐 맵을 압축해서 서버로 전송해야 한다. 하지만 피쳐 맵의 압축으로 인해 정확도(accuracy)의 손실이 발생한다. 따라서 정확도의 손실을 최소화하는 압축기법이 필요하다. JPEG과 같은 일반적인 압축기법은 지연시간을 줄일 수 있지만, 정확도 손실이 크다.

그리고 압축을 할 때는 현재의 네트워크 상태도 고려할 필요가 있다. 네트워크 상태가 좋지 않다면 압축을 많이 해서 지연시간을 줄여야 하고 네트워크 상태가 좋다면 압축을 조금만 해서 정확도를 올려야 한다. 이것이 가능하도록 압축률이 동적으로 조절 가능한 압축기법이 필요하다.

딥 러닝 모델에서 압축하는 방법에는 피쳐 맵을 프루닝 하는 것이 있다. 피쳐 맵을 프루닝 하는 연구에는 채널의 중요도를 판단해 Top-k 방식으로 프루닝을 하는 연구[3] 등이 있다. [3]의 연구에서 사용하는 방식은 프루닝을 할 비율을 정해놓고 한 가지의 프루닝 비율로만 프루닝을 진행한다. 따라서 이러한 모델을 동적으로 프루닝 비율을 조절할 수 있도록 하려면 여러 프루닝 비율을 사용해서 joint training을 시켜주어야 한다.

joint training은 여러 개의 loss를 더해서 하나의 최종 loss로 학습을 시키는 방식이다. joint training을 사용해서 학습을 시킨 연구에는 full 모델을 teacher 모델로 사용하고 프루닝 한 모델을 student 모델로 사용해서 in-place distillation도 사용해 joint training을 시킨 연구[4] 등이 있다.

본 논문에서는 압축률을 동적으로 조절할 수 있고 압축에 따른 정확도 손실이 적은 프루닝 기법을 소개한다.

이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.IITP-2017-0-00914, 지능형 IoT 장치용 소프트웨어 프레임워크)

2. 배경 지식 및 관련 연구

2.1 슬라이싱(slicing) 및 Top-k 프루닝(pruning)

피쳐 맵의 압축을 위해 사용하는 기법으로는 슬라이싱과 Top-k 프루닝이 있다.

슬라이싱 기법은 그림 1(a)와 같이 피쳐 맵을 일정 비율로 잘라내는 기법이다. 슬라이싱 기법은 일정 비율에 해당하는 부분을 채널의 중요도에 상관없이 연속적으로 자른다. 따라서 중요도가 높은 채널들이 잘리는 경우가 발생할 수 있다. 이는 정확도 손실을 발생시키는 원인이 될 수 있다. 하지만 연속적으로 채널을 잘라내기 때문에 하드웨어에 친화적이라는 장점이 있다. 슬라이싱과 관련된 기존연구 중 네트워크 상태에 따라서 채널을 슬라이싱하는 연구[1]가 있다. 연구에서는 네트워크 상태에 따라서 슬라이싱하는 비율을 다르게 설정한다.

Top-k 프루닝 기법은 그림 1(b)와 같이 피쳐 맵 에서 중요하지 않은 채널을 선택적으로 잘라내는 기법이다. 프루닝 기법은 중요한 채널과 중요하지 않은 채널을 구분해서 잘라내기 때문에 슬라이싱 기법보다 정확도 손실이 적다. 하지만 채널을 연속적으로 잘라내지 않기 때문에 하드웨어에 친화적이지 않다는 단점이 있다. Top-k 프루닝을 하는 연구에는 채널들의 중요도 순위에 따라 프루닝을 하는 연구[3] 등 여러 가지가 있다.

3. 슬라이싱 + Top-k 프루닝 기법

이 연구에서는 슬라이싱 방법과 Top-k 프루닝 방법을 모두 사용한 새로운 프루닝 기법을 제안한다.

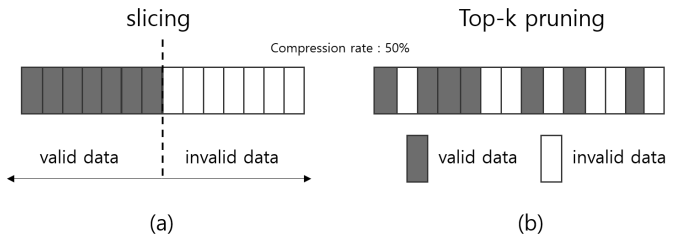

그림 1 슬라이싱 기법과 프루닝 기법. (a) 슬라이싱 기법을 나타낸다. 피처 맵의 일정 비율을 연속적으로 잘라낸다. (b) Top-k 프루닝 기법을 나타낸다. 피처 맵의 중요도가 낮은 채널을 잘라낸다.

정확도 손실의 최소화를 위해 Top-k 프루닝 기법만을 사용했을 때 모델의 학습이 잘되지 않는 것을 실험적으로 발견했다. Top-k 프루닝 기법은 피쳐 맵 에서 중요한 채널을 남기고 중요하지 않은 채널들을 잘라낸다. 잘린 채널에 따라서 학습되는 weight 필터도 달라진다. Top-k 프루닝의 경우 입력마다 선택되는 채널이 달라지고 따라서 학습되는 weight 필터도 매번 달라진다. 이러한 이유로 다양한 프루닝 비율로 모델을 joint training을 시켰을 때 학습이 제대로 되지 않는다. Top-k 프루닝 방식은 높은 압축률 일 경우는 중요한 채널을 선택하기 때문에 정확도가 높다. 하지만 학습이 제대로 되지 않기 때문에 압축을 하지 않은 full 모델을 사용했을 때와 낮은 압축률로 압축을 했을 때, 슬라이싱 기법만을 사용한 기존 방법[1]보다 정확도가 낮다.

따라서 본 논문에서 사용할 프루닝 기법은 입력마다 선택되는 채널이 일정한 슬라이싱 기법과 Top-k 프루닝 기법을 모두 사용해서 학습이 잘 될 수 있도록 한다.

그림 2(a)와 같이 피쳐 맵의 90%(10배)를 프루닝 할 때까지는 슬라이싱 기법을 사용하고, 그림 2(b)와 같이 90%보다 더 많이 프루닝을 할 때는 슬라이싱 기법과 Top-k 프루닝 기법을 같이 사용한다. 슬라이싱과 Top-k 프루닝 기법을 사용하는 비율은 실험적으로 선택했다.

동적으로 달라질 수 있는 압축률로 학습시키기 위해 여러 모델을 joint training 시키고, knowledge distillation을 사용한다. 학습을 시키기 위한 loss를 구할 때 두 가지의 loss를 사용한다.

첫 번째 loss는 그림 3의 (1)과 같이 label과 student 모델들의 output을 사용해서 loss를 구한다. 각 student 모델들은 압축률을 다르게 설정한 모델들이다. student i까지 loss를 전부 구한 후 모든 loss를 더한다.

두 번째 loss는 그림 3의 (2)와 같이 teacher 모델(pre-trained 모델)의 output과 student 모델의 output을 사용해서 loss를 구한다. 첫 번째 loss와 마찬가지로 student i 까지 loss를 전부 구한 후 모든 loss를 더한다. 첫 번째 loss와 두 번째 loss를 식으로 나타내면 아래와 같다.

$$Loss_1 = \sum_{i=1}^{N} CE(\sigma(M_i(x)), y) \quad (1)$$

$$Loss_2 = \sum_{i=1}^{N} CE(\sigma(M_i(x)), \sigma(M(x))) \quad (2)$$

CE는 Cross Entropy loss, σ는 softmax function, $M_i(x)$는 student 모델의 결과(output), M(x)는 teacher 모델의 결과, y는 label, N은 student 모델의 수를 의미한다.

training에 사용할 최종 loss는 첫 번째 loss와 두 번째 loss를 더해서 사용한다. 따라서 최종 loss는 아래와 같이 나타낼 수 있다.

$$Loss_{final} = \alpha \sum_{i=1}^{N} CE(\sigma(M_i(x)), y) + (1-\alpha) \sum_{i=1}^{N} CE(\sigma(M_i(x)), \sigma(M(x))) \quad (3)$$

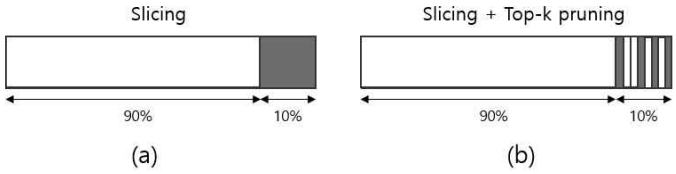

그림 2 (a) 슬라이싱 기법을 나타낸다. 90%(10배) 이하의 압축률에서 피처 맵의 일정 비율을 연속적으로 잘라낸다. (b) 슬라이싱 + Top-k 프루닝 기법을 나타낸다. 90% 이상의 압축률에서 90%까지 슬라이싱을 하고 남은 10%의 피처 맵의 중요도가 낮은 채널을 잘라낸다.

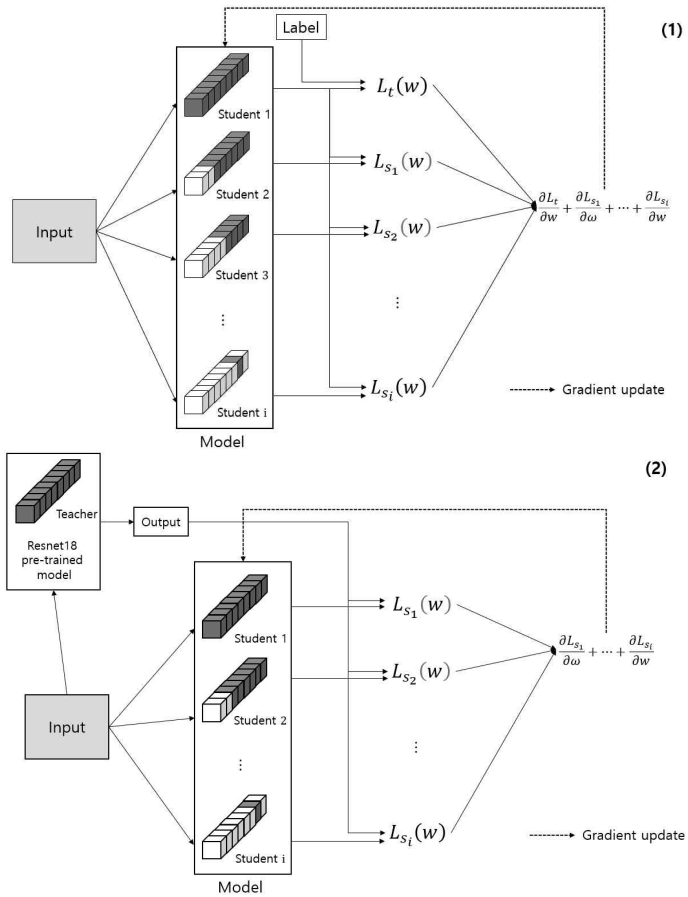

그림 3 모델 학습 예시. (1)에서는 Label과 student의 output을 사용해서 loss를 구하고 (2)에서는 Teacher model(pre-trained model)의 output과 student의 output을 사용해서 loss를 구한다. 최종적으로 (1)과 (2)의 loss를 더한 값을 학습 때 사용한다.

최종 loss에서 사용한 α는 $Loss_1$과 $Loss_2$의 비율을 조절하기 위한 파라미터이다. 위의 knowledge distillation 방법은 label과 pre-trained 모델을 사용하는 기존의 연구[5]와 유사하다.

다양한 압축률에 대해서 학습을 시키기 위해 매 epoch마다 최소 압축률과 최대 압축률을 정하고 최소 압축률과 최대 압축률 사이의 임의의 두 압축률을 선정해 총 4가지의 압축률을 사용해 학습을 시킨다. 이는 매 epoch마다 4가지의 student 모델을 사용해 학습을 시키는 것이다. 이 방법은 US-Net[4]에서 사용한 sandwich rule 방식이다.

4. 실험 환경 및 결과

4.1 실험환경

실험에서 사용된 DNN 모델은 ResNet18 이고, 데이터 셋으로

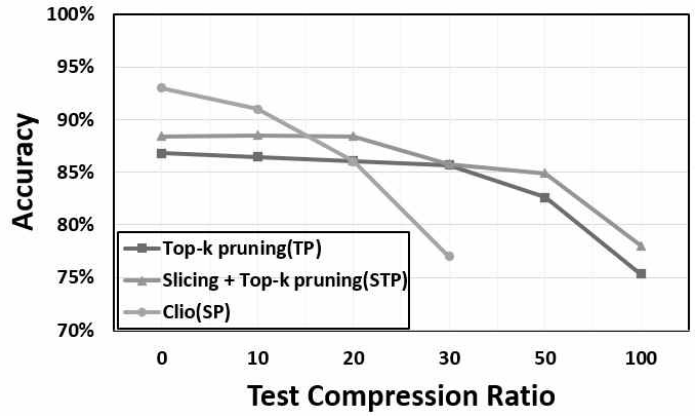

그림4 슬라이싱 + Top-k 프루닝 기법으로 학습된 모델의 정확도 측정

CIFAR-10을 사용했다.

실험 파라미터 설정으로는 pre-trained 모델을 teacher 모델로 사용했고 Compression Ratio(CR)가 0, 95%(20배)인 경우를 각각 최소, 최대 CR로 사용했다. 그리고 최소, 최대 CR 사이의 임의의 두 가지 CR를 선정해 총 4가지의 CR의 모델을 student 모델로 사용했다. student 모델들을 joint training하고 학습된 모델을 CR가 0배인 경우부터 100배일 경우까지 테스트했다.

4.2 실험결과

그림 4는 CR에 따른 Slicing + Top-k 프루닝(STP)을 사용해 학습시킨 모델의 결과, Clio[1] 논문(SP)의 결과 그리고 Top-k 프루닝 방식(TP)만 사용했을 때의 결과를 보여준다. Clio[1] 논문에서 사용한 방식은 Slicing만을 사용한 방식이다.

그림 4에서 압축을 하지 않았을 경우, 10배 압축을 했을 경우 SP의 결과는 각각 93%, 91% 정도로 높은 정확도를 보여준다. 하지만 압축률이 20배일 경우 86% 정도의 정확도로 정확도 손실이 발생하고 압축률이 올라갈수록 정확도 손실이 매우 크다.

반면 압축을 하지 않았을 경우와 10배 압축을 했을 경우 STP의 결과와 TP의 결과는 각각 86%, 88% 정도로 SP의 결과보다 낮다. 20배 이상의 압축률에서 TP의 경우 86%로 SP와 비슷하고 STP의 경우 정확도가 88% 정도로 SP와 TP의 결과보다 높은 정확도가 나온다. STP의 경우 50배의 높은 압축률로 압축을 했을 때도 85%의 정확도를 보여준다.

4.3 실험 분석

그림 4의 결과를 보면 TP의 결과보다 STP의 결과가 더 높게 나온 것을 볼 수 있다.

TP의 경우 joint training을 할 때 들어온 입력에 따라서 선택되는 채널이 매번 달라진다. 선택되는 채널이 달라지면 학습되는 weight 필터도 달라진다. 따라서 joint training을 시킬 때 입력에 따라 매번 학습되는 weight 필터가 달라지고 압축률에 따라서 학습되는 weight 필터의 수도 달라진다. 따라서 weight 필터들이 불규칙하게 학습이 되고 따라서 전체 모델의 학습이 잘 되지 않는다.

반면 STP의 경우 피처 맵의 90%까지는 슬라이싱 방식을 사용하고 10%만 Top-k 방식으로 학습을 시키기 때문에 90%의 까지는 일정하게 피처 맵의 채널이 선택되고 그에 따른 weight 필터도 일정하게 학습이 된다. 그리고 10% 정도의 피처 맵만 불규칙하게 선택이 되기 때문에 불규칙하게 학습되는 weight 필터의 비율 또한 적다. 따라서 TP 방식보다 STP 방식이 학습이 더 잘 된다.

그림 5는 다섯 개의 입력이 들어왔을 때 TP의 경우 실제로 선택되는 채널의 인덱스를 보여준다. 선택된 채널 인덱스를 보면 입력마다 매번 선택되는 채널들도 있고 매번 선택되진 않지만 자주 선택되는 채널들도 있다. 이런 채널들을 제외한 다른 채널들의 경우에는 입력마다 무작위로 채널이 선택된다.

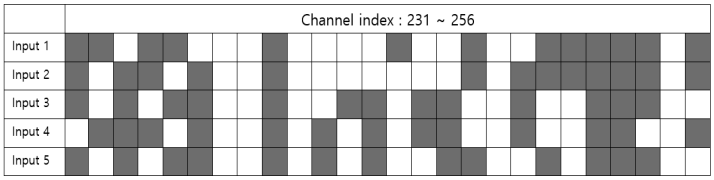

그림5 Top-k 프루닝 방식을 사용했을 때 실제 선택되는 채널의 인덱스

그리고 그림 5는 채널 인덱스가 231부터 256인 경우만 보여주고 있는데 모든 채널의 경우 선택되는 채널의 인덱스는 더욱 무작위로 채널이 선택된다.

5. 결론 및 향후 연구

본 논문에서는 다양한 압축률로 동작하고 압축에 따른 정확도의 손실이 적은 프루닝 기법을 제안하였다. 다양한 압축률로 동작할 수 있도록 joint training 기법과 knowledge distillation 기법을 사용해 모델을 학습시켰다.

다양한 압축률로 학습된 모델은 최대 50배 압축을 시켰을 때도 압축을 하지 않았을 때보다 정확도 손실이 8% 이하였다.

슬라이싱 + Top-k 프루닝 기법을 사용해서 학습된 모델은 20배 이상의 높은 압축률에서의 정확도가 높다. 하지만 압축을 하지 않았을 경우와 낮은 압축률에서는 정확도가 기존 슬라이싱만 사용해서 학습된 연구[1]보다 낮게 나온다. 따라서 낮은 압축률에서도 슬라이싱 + Top-k 프루닝 기법의 정확도가 높아지도록 학습시키는 것이 향후 연구할 과제이다.

그리고 게이트웨이 오프로딩 시 네트워크 상태 등 동적으로 변하는 요소들에 의해 모바일 장치에서 클라우드 서버로 데이터를 전송할 때 발생하는 지연시간이 달라진다. 따라서 동적으로 변하는 요소들이 미치는 영향을 최소화할 필요가 있다. 이를 위해 다양한 압축률로 동작할 수 있도록 학습된 모델을 실제 딥 러닝 모델의 연산을 모바일 장치와 서버에 분할시키는 오프로딩에 적용하는 것도 향후 연구할 과제이다.

6. 참고문헌

[1] J. Huang, C. Samplawski, D. Ganesan, B Marlin, and H. Kwon. 2020. Clio: Enabling automatic compilation of deep learning pipelines across IoT and Cloud. In The 26th Annual International Conference on Mobile Computing and Networking (Mobicom).

[2] Stefanos Laskaridis, Stylianos I. Venieris, Mario Almeida, Ilias Leontiadis, and Nicholas D. Lane. SPINN: Synergistic Progressive Inference of Neural Networks over Device and Cloud. In International Conference on Mobile Computing and Networking (MobiCom), 2020.

[3] Xitong Gao, Yiren Zhao, Lukasz Dudziak, Robert Mullins, Cheng-zhong Xu. 2019. Dynamic Channel Pruning: Feature Boosting And Suppression. In International Conference on Learning Representation (ICLR).

[4] Jiahui Yu, Thomas Huang. 2019. Universally Slimmable Networks and Improved Training Techniques. In Proceeding of the IEEE/CVF International Conference on Computer Vision (ICCV).

[5] Frederick Tung, Greg Mori, 2019, Similarity-Preserving Knowledge Distillation. In Proceedings of the IEEE/CVF International conference on Computer Vision (ICCV).