JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 22, No. 9: 1165~1171, Sep. 2018

지역적 가중치 파라미터 제거를 적용한 CNN 모델 압축

임수창[1] · 김도연[1*]

Apply Locally Weight Parameter Elimination for CNN Model Compression

Su-chang Lim[1] · Do-yeon Kim[1*]

[1*]Department of Computer Engineering, Sunchon National University, Sunchon 57922, Korea

요 약

CNN은 객체의 특징을 추출하는 과정에서 많은 계산량과 메모리를 요구하고 있다. 또한 사용자에 의해 네트워크가 고정되어 학습되기 때문에 학습 도중에 네트워크의 형태를 수정할 수 없다는 것과 컴퓨팅 자원이 부족한 모바일 디바이스에서 사용하기 어렵다는 단점이 있다. 이러한 문제점들을 해결하기 위해, 우리는 사전 학습된 가중치 파일에 가지치기 방법을 적용하여 연산량과 메모리 요구량을 줄이고자 한다. 이 방법은 3단계로 이루어져 있다. 먼저, 기존에 학습된 네트워크 파일의 모든 가중치를 각 계층 별로 불러온다. 두 번째로, 각 계층의 가중치에 절댓값을 취한 후 평균을 구한다. 평균을 임계값으로 설정한 뒤, 임계 값 이하 가중치를 제거한다. 마지막으로 가지치기 방법을 적용한 네트워크 파일을 재학습한다. 우리는 LeNet-5와 AlexNet을 대상으로 실험을 하였으며, LeNet-5에서 31x, AlexNet에서 12x의 압축률을 달성 하였다.

ABSTRACT

CNN requires a large amount of computation and memory in the process of extracting the feature of the object. Also, It is trained from the network that the user has configured, and because the structure of the network is fixed, it can not be modified during training and it is also difficult to use it in a mobile device with low computing power. To solve these problems, we apply a pruning method to the pre-trained weight file to reduce computation and memory requirements. This method consists of three steps. First, all the weights of the pre-trained network file are retrieved for each layer. Second, take an absolute value for the weight of each layer and obtain the average. After setting the average to a threshold, remove the weight below the threshold. Finally, the network file applied the pruning method is re-trained. We experimented with LeNet-5 and AlexNet, achieved 31x on LeNet-5 and 12x on AlexNet.

키워드 : CNN, 가지치기, 가중치 압축, 모델 압축

Key word : CNN, Pruning, Parameter Compression, Model Compression

Received 28 May 2018, **Revised** 19 July 2018, **Accepted** 21 Aug 2018

*** Corresponding Author** Do-Yeon Kim(E-mail:dykim@scnu.ac.kr, Tel:+82-61-750-3628)
Department of Computer Engineering, Sunchon National University, Sunchon 57922, Korea

Open Access http://doi.org/10.6109/jkiice.2018.22.9.1165 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

최근 딥러닝은 이미지 인식[1-2], 객체 검출과 같은 컴퓨터 비전 작업에서 음성인식, 자연어 처리까지 다양한 분야에서 사용되고 있다. 특히, 딥러닝 알고리즘중 하나인 CNN (Convolutional Neural Network) [3]은 ILSVRC (ImageNet Large Scale Visual Recognition Challenge)[4] 에서 우수한 분류 성능을 달성하며 이미지 인식 분야에서 사용되는 대표 신경망으로 자리 잡고 있다. CNN은 학습 데이터로부터 특징을 추출하고 학습하기 위해 다층의 컨볼루션 레이어와 풀리 커넥티드 레이어가 결합된 구조로 이루어져 있다. CNN의 분류 성능은 다수의 레이어가 사용된 심층망 형태로 제작될 때 향상된다고 알려져 있다. CNN은 특징 추출에 사용되는 커널의 형태, 개수에 따라 많은 가중치 파라미터를 포함한다. 이는 망 전체의 계산 복잡도와 메모리 사용량을 증가 시키는 주요 원인이다. 2012년 ILSVRC에서 기존 기계학습 방법들 보다 우수한 성능을 보여주며 우승한 AlexNet[5]은 5개의 컨볼루션 레이어, 3개의 풀리커넥티드 레이어로 구성되어 있다. 이 네트워크는 60M 이상의 파라미터를 포함하는 방대한 구조로 인해 GPU를 사용하여 학습하였다. 이와 같은 구조는 연산 과정에서 많은 계산 능력과 메모리를 요구하기 때문에 소형 임베디드 모듈 및 모바일 디바이스와 같이 성능이 제한된 환경에 적용하기 어렵다. 우리는 60M개 이상의 파라미터 중에는 특징 추출에 미미한 영향을 끼치는 파라미터도 존재할 것이라고 가정하였다. 따라서, 이 가정을 기준으로 위에서 언급한 제약사항을 해결하기 위해서 우리는 기존 네트워크의 정확도를 유지하며 네트워크에서 요구하는 많은 계산량과 메모리 양을 줄이기 위해 분류 성능에 미미한 영향을 미치는 가중치를 삭제하여 압축하는 가지치기(Pruning) 방법을 제안한다. 첫째로, 사전 학습된 네트워크 모델을 로드한다. 이러한 이유는 CNN은 초기화때 설계한 네트워크 구조가 고정되어있기 때문에 학습 도중에 네트워크 구조를 수정할 수 없기 때문이다. 두 번째로, 로드된 네트워크 모델의 각 네트워크에 압축 비율을 적용하여 가중치를 삭제한다. 이때 첫 번째 레이어는 색상 및 엣지 등 의 특징을 추출하는 중요한 역할을 지니고 있기 때문에 압축 비율은 낮게 지정하고, 분류에 사용되는 풀리 커넥티드 레이어는 전체 파라미터 중 절반 이상을 포함하기 때문에 가장 높은 압축비율을 지정하여 가중치를 삭제한다. 마지막으로 가중치 최적화를 위해 압축된 네트워크 모델을 재학습한다. 이에 본 논문에서는 네트워크 압축을 위한 가지치기 알고리즘을 적용하여 압축률에 따른 분류 정확도를 관찰하고 분석하였다. 방법을 적용할 CNN으로는 LeNet-5, AlexNet을 선택하였다. 논문의 구성은 다음과 같다. 섹션 2에서는 관련 연구에 대하여 기술하였고, 섹션 3에서는 제안하는 가지치기 방법에 대한 세부 내용을 기술하였다. 섹션 4에서는 실험 및 결과에 대하여 기술하였으며, 마지막 섹션 5에서는 논문을 결론 짓는다.

Ⅱ. 관련연구

CNN 은 컨볼루션 필터의 가중치를 통해 입력 데이터의 특징을 추출한다. 이때 가중치 개수가 많을수록 계산량과 사용되는 메모리가 많아지므로 높은 하드웨어 성능이 요구된다. 이러한 방법을 해결하기 위해 다양한 알고리즘을 적용한 방법들이 있다. Gong et al.[6]은 CNN 압축을 위해 기존 행렬 인수 분해 방법보다 더 효과적인 백터 양자화를 이용하였다. 이 방법을 적용함으로써 네트워크를 16-24배로 압축하였다. Vanhoucke et al.[7]은 기존 32비트 부동소수점으로 저장되는 파라미터를 8비트 정수형태로 바꿔 저장함으로써 모델의 용량을 줄였다. Chandrasekhar et al.[8]은 인스턴스 검색 문제를 해결하기 위해 양자화, 코딩, 가지치기, 그리고 가중치 공유 기술을 통해 모델의 크기를 줄였다. 다른 방법으로, Chen et al.[9]은 해시 함수를 사용하여 연결된 가중치를 해시 버킷으로 무작위 그룹화하여 모델 크기를 줄였다. 이 방법은 해시 버킷 내의 모든 연결이 단일 매개 변수 값으로 공유된다. 가중치 압축 방법 이외에 네트워크의 구조를 변경함으로써 네트워크 압축을 시도한 연구 결과도 있었다. Lin et al.[10]과 Szegedy et al.[11]은 풀리 커넥티드 레이어를 전역 평균 풀링으로 변경하여 네트워크의 매개변수를 줄였다. S Srinivas et al.[12]는 데이터 계산을 위한 각 레이어중 풀리 커넥티드 레이어(Fully Connected Layer)의 뉴런을 대상으로 ReLu 활성함수를 변형한 추가 매개변수 곱을 통해 1과 0의 이진값을 출력하여 0으로 출력된 뉴런을 찾아 제거하는 네트워크 압축 알고리즘 방법을 제안하였다. 그러나 이미지 인식을 위해 사용되는 풀리 커넥티드 레이어와 컨볼루

션 레이어를 비교할 때, 컨볼루션 레이어가 계산적비중이 더 크며[13], 또한 풀리 커넥티드 레이어만을 대상으로한 네트워크 압축 방법은 구현의 어려움이 있는 단점이 있다.

Ⅲ. 가중치 압축

CNN은 대표적으로 컨볼루션 레이어(Convolution Layer), 풀리 커넥티드 레이어(Fully Connected Layer)로 구성되어있으며, 선택적으로 풀링 레이어와 활성화 함수 역할을 하는 ReLU(Rectified Linear Unit) 레이어를 추가할 수 있다. 이때 컨볼루션 레이어에서 특징을 추출하는 Convolution연산으로 인해 많은 계산량이 발생하고 풀리 커넥티드 레이어는 가장 많은 파라미터를 포함하기 때문에 요구되는 메모리가 가장 많다. CNN이 지닌 특징은 수백만개의 파라미터 중 중복값이 존재하는 것과 가중치가 0에 가까운 값일수록 행렬연산 결과에 큰 영향을 미치지 못한다는 것이다. 따라서 비교적 중요하지 않은 가중치를 0으로 바꾸는 것을 가지치기(Pruning)이라 한다. 이를 통해 연산 속도를 향상시키고 메모리 양을 줄일 수 있다. 본 논문에서는 특정 임계값보다 작은 가중치의 압축 비율을 조절하며 가중치를 0으로 바꾸기 위한 가지치기 알고리즘을 제안한다.

가지치기 알고리즘은 그림 1처럼 3단계로 적용된다. 첫 번째로, 가지치기 알고리즘을 적용할 사전 학습된 네트워크 학습 모델을 불러온다. 이는 학습 도중에 모든 가중치가 지속적으로 변하기 때문에 어떤 가중치가 성능에 미미한 영향을 미치는지 알 수 없어, 학습이 완료된 가중치를 사용할 목적이다. 두 번째로, 로드된 네트워크 모델에서 각 레이어의 특징 추출 중요도와 밀집된 가중치의 양에 따라 가중치를 0으로 변환한다. 우선, 각 레이어의 가지치기 비율을 설정해야한다. 여기서 비율은 사용자가 임의로 설정하는 값으로 몇 개의 가중치에 가지치기를 적용할 것인지를 결정한다. 다음으로, 가지치기 기준이 되는 임계값은 레이어별로 다음과 같이 획득된다. 임계값을 결정하는 각 레이어의 가중치는 양수 및 음수 값으로 이루어져 있기 때문에 우선 절대 값으로 변환한 후 이를 합산하여 평균을 구하고 이를 임계값으로 사용한다. 이후, 각 레이어의 가중치를 오름차순으로 정렬한 뒤, 사전에 설정한 압축 비율에 따라 임계값 미만

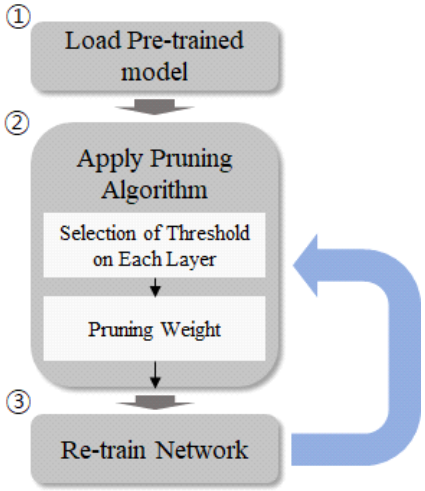

Fig. 1 Pruning algorithm flow chart

인 가중치를 0으로 변환 시킨다. 가지치기 이후 잔여 가중치는 원본으로 교체한다. 그림 2는 가지치기를 적용한 컨볼루션 레이어이다. 흰색 점선은 가지치기된 가중치로서 0값이 되어 다음 레이어와의 연결이 단선되었다.

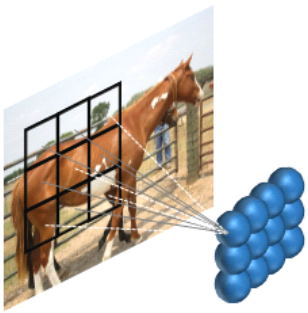

Fig. 2 Neurons in the convolution layer after pruning

가지치기가 적용된 풀리 커넥티드 레이어 노드의 형태는 임계값 미만의 가중치가 0으로 치환되므로, 이 가중치 값과 관계된 모든 연결이 네트워크에서 제거된다. 또한 가중치 밀집도가 높은 레이어는 스파스 네트워크 형태로 변환된다. 마지막으로 살아남은 뉴런을 최적화하기 위해 재훈련 단계를 거친다. 기존 사전 학습된 모델은 ImageNet 1000 classes 학습데이터의 특징 추출에 최적화 되어있지만 푸르닝 알고리즘을 적용함으로써 구조가 붕괴 되었다. 따라서 압축된 상태로 분류 문제에 사용할 경우 10%미만의 분류 결과를 보여준다. 이러한 문제점은 각 가중치가 0이 아닌 값을 재학습함으로써

해결할 수 있다. 재훈련 단계에서 역전파 과정을 거치며 죽은 뉴런의 입력 연결이 제로이고 출력 연결이 제로가 됨으로 최적화에 자동으로 도달한다. 이는 입력 연결이 0 인 뉴런 (또는 출력 연결이 0 인 뉴런)은 최종 손실값에 아무런 영향을 미치지 않으므로 출력 연결 (또는 입력 연결)에 대해 각각 그라디언트가 0이 된다. 따라서 역전파 과정 중에 출력에 영향을 미치지 않는 제로 가중치가 전파되며 살아남은 가중치만 학습이 이루어진다. 이 과정에서 죽은 신경 세포는 재훈련 중에 자동으로 제거된다.

Ⅳ. 실 험

우리는 카페 프레임워크[14]를 이용해 네트워크 가지치기 및 재학습을 시행했다. 카페 프레임워크는 가지치기 작업과 zero weight 되살리기를 방지하기 위해 수정되었다.

```
[[[ 0.04475803  0.0369841  -0.03168138 -0.08679054 -0.0917629 ]
  [ 0.01770928 -0.0139185   0.03713696  0.00715715 -0.03922879]
  [ 0.04057263  0.07790662  0.11699135  0.07796392  0.06770425]
  [ 0.03791948 -0.00553014  0.02236695  0.08733234  0.05307791]
  [-0.05869114 -0.05191451 -0.05290764 -0.03805342  0.05680364]]]


[[[ 0.04013258  0.03035274  0.02197575 -0.07736494 -0.08612953]
  [-0.01690165  0.02636294 -0.06355806 -0.02626047 -0.04941376]
  [ 0.03671333  0.05990707 -0.03570092  0.01506261  0.06961753]
  [ 0.07831739  0.08781832  0.09352255  0.07214608  0.01666274]
  [ 0.0542119   0.0543012   0.05728562  0.00501869 -0.00340483]]]]
```

Fig. 3 Pruning algorithm result before the original source code modification

원본 카페 프레임워크는 그림 3처럼 재학습 과정 중에 죽은 뉴런이 다시 살아나는 현상이 나타났다. 이는 가중치 수를 줄이지 못하며 압축의 의미가 사라지기 때문이다. 따라서 재학습 중에 zero weight를 동결시키고 나머지 노드간의 연결을 재정의 하며 가중치를 학습하기 위해 소스코드를 수정하였다. 그림 4는 가중치 가지치기가 적용된 커널이다.

본 논문에서는 MNIST 데이터셋을 사용한 LeNet-5와 ImageNet 데이터셋을 사용한 AlexNet을 대상으로 가지치기 알고리즘을 적용해 네트워크 압축을 진행하였으며, 인텔 I7-7700K 그리고 Nvidia TitanXp로 실험을 수행했다.

```
[[[ 0.          0.          0.          0.          0.        ]
  [ 0.         -0.05499713  0.          0.          0.        ]
  [ 0.          0.          0.          0.05999395  0.        ]
  [ 0.          0.05696519  0.          0.          0.        ]
  [ 0.          0.          0.          0.          0.        ]]]


[[[ 0.          0.          0.          0.          0.        ]
  [ 0.          0.          0.          0.          0.        ]
  [ 0.          0.          0.          0.          0.        ]
  [ 0.          0.          0.          0.06033171  0.        ]
  [ 0.          0.          0.          0.05568442  0.        ]]]
```

Fig. 4 Kernel type with weighted pruning

4.1. LeNet-5 네트워크 압축 결과

LeNet-5[15] 네트워크 모델은 2개의 컨볼루션 레이어(conv)와 2개의 풀리 커넥티드 레이어(fc)로 구성되어 있다.

Table. 1 Each layer weight parameter of the LeNet-5 network with pruning algorithm

Layers	Number of Parameter		
	Original Parameter	Compressed Parameter	Compression Rate
Conv1	500	64	8x
Conv2	25,000	1,248	20x
Fc1	400,000	11,999	33x
Fc2	5,000	401	12x
Total	430,500	13,712	31x

표 1은 LeNet-5 네트워크 모델을 대상으로 가지치기 알고리즘이 적용된 각 레이어의 압축 성능 결과이다. LeNet-5 네트워크에 원본 가중치 개수 대비 conv1 : 88%, conv2 : 95%, fc1 : 97%, fc2 : 92%로 압축 범위를 지정하고 가중치 평균값을 이용해 임계값을 설정하였다. 실험에서 기존 네트워크의 430,000개 파라미터와 비교했을때, 13,712개로 줄이며 약 31x의 파라미터 압축을 보였으며, 분류 정확률 손실은 0.04%에 그쳤다. 우리는 동일한 MNIST 데이터셋을 이용한 LeNet-5 네트워크 압축 연구와 제안한 알고리즘을 비교하며 성능을 검증하였다. SVD(rank-10)[16] 알고리즘은 커널을 대각 행렬로 변환하여 노드간의 연결 최소화를 통해 계산 속도 향상 및 네트워크 압축 연구를 진행하였으며, Architecture Learning[12]은 ReLU 함수를 변형해 매개변수를 추가한 뒤, 이를 사용해 재학습이 필요 없이 학습을 진행하는 동안 네트워크를 압축하는 방법을 제시하였다. Fastfood-1024[17] 알고리즘은 커널의 대각행렬 변환과 하다마르 행렬 등을 통해 네트워크 압축을 진

행하였으며, S han[18]은 표준 편차를 이용해 임계 값을 구해 네트워크 압축을 진행하였다. 성능 비교 결과 본 연구에서 제시한 압축 알고리즘 방법으로 LeNet-5 네트워크 파라미터를 31x를 줄이며 최종적으로 0.04%의 분류 정확도가 감소하는 결과를 얻으며 가장 우수한 압축률을 보였다. 표 2는 LeNet-5 네트워크의 압축 성능비교 표이다.

Table. 2 LeNet-5 Network Compression Performance Comparison

Method	Result		
	Number of Parameter	Accuracy(%)	Compression
LeNet-5[15]	431K	99.11	1x
SVD (rank-10) [16]	43.6K	98.47	10x
Architecture Learning[12]	40.9K	99.04	10.5x
Fastfood-1024 [17]	38.8K	99.29	11x
S han et al.[18]	36K	99.23	12x
Proposed Method	13.7K	99.09	31x

4.2. AlexNet 네트워크 압축 결과

AlexNet은 5개 컨볼루션 레이어와 3개의 풀리 커넥티드 레이어로 구성되어있다. 그림 5는 이 네트워크가 지닌 가중치 파라미터 수이다. fc1에 가중치 파라미터가 집중되어 있다는 것을 알 수 있다.

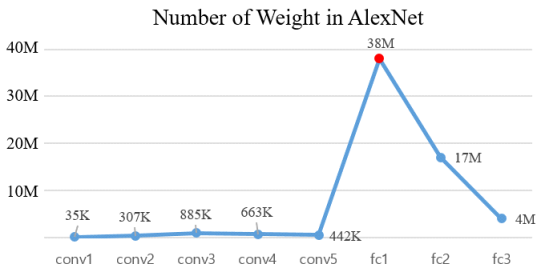

Fig. 5 The number of weight parameter of AlexNet

우리는 동일한 가지치기 방법을 AlexNet에 적용하여 네트워크를 압축 하였다. 각 레이어에 적용할 압축률을 선정할 때 우리는 LeNet-5와 다른 관점에서 접근했다.

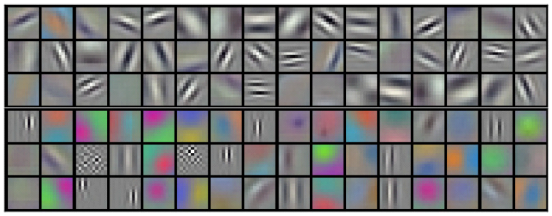

Fig. 6 96 convolutional kernels of size 11×11×3 learned by the first convolutional layer on the 224×224×3 input images.[5]

LeNet-5와 AlexNet의 차이점은 분류대상의 속성이 다르다. LeNet-5은 그레이스케일 영상을 분류하는데 특화되어 있으며, AlexNet은 칼라 영상을 분류하는데 이용된다. LeNet-5의 첫 번째 컨볼루션 레이어에서 사용되는 필터의 깊이는 1채널이다. 이는 단순히 추출하고자 하는 객체의 색상을 고려하지 않고 경계를 추출하는데 특화되어 있는 것을 알 수 있다. 하지만 알렉스 넷은 첫 번째 컨볼루션 레이어에서 색상 정보를 추출하기 위해 필터의 깊이는 3채널로 구성되어 있으며 각 필터는 그림 6처럼 단순 엣지부터 색상 정보까지 추출하는 다양한 형태로 학습된다. 이 특성을 고려하여 우리는 칼라 특징을 추출하는 첫 번째 컨볼루션 레이어의 역할이 중요하다고 판단하고 이 레이어의 압축률을 낮게 설정하였다.

Table. 3 Weight Parameter of AlexNet with Pruning Algorithm

Layers	Number of Parameter		
	Original Parameter	Compressed Parameter	Compress Rate
conv1	35K	29K	1.2x
conv2	307K	116K	2.6x
conv3	885K	309K	2.9x
conv4	663K	232K	2.9x
conv5	442K	154K	2.9x
fc1	38M	2.6M	14.6x
fc2	17M	1.3M	13.1x
fc3	4M	1M	4x
Total	61M	5M	12.2x

표3은 AlexNet모델에 가지치기 알고리즘을 적용한 결과이다. conv1은 칼라 특징을 추출해야하는 중요한 레이어 이므로 압축률을 16%로 가장 낮게 적용하였으며,

가장 많은 파라미터를 지닌 풀리 커넥티드 레이어는 90% 이상 압축하였다. 우리는 표4와 같이 ImageNet ILSVRC 2012 데이터세트를 사용하는 AlexNet 압축 방법 연구 결과와 제안한 가지치기 알고리즘 결과를 비교하여 성능 검증 하였다.

Table. 4 Compression performance comparison of AlexNet

Method	Result		
	Number of Parameter	Accuracy(%)	Compression
AlexNet[5]	61M	57.22	1x
Data-free Pruning[19]	39.6M	55.6	1.5x
Fastfood-32-AD [17]	32.8M	58.07	2x
Collins &Kohli [20]	15.2M	55.6	4x
SVD[16]	11.9M	55.98	5x
S Han et al[18]	6.7M	57.23	9x
Proposed Method	5M	56.2	12x

Data-free pruning[19]은 오직 1.5x 압축했음에도 불구하고 2%이상 정확도 손실이 발생하였다. Fastfood-32-AD[17]는 풀리 커넥티드 레이어를 대상으로 압축을 진행하였으며 파라미터를 원본과 비교해서 2x 줄이며 정확도를 약 0.8% 향상시켰다. Collins & Kohli[20]은 파라미터를 4x 줄인 반면 [17]와 비교했을 때 더 낮은 정확도를 보여주었다. SVD[16]는 convnet의 선형 구조를 이용하여 각 레이어를 개별적으로 압축하며 5x의 압축률을 보여주었다. S Han et al[18]은 표준편차를 이용해 획득한 임계 값을 이용하여 각 레이어를 압축하였으며, 원본과 동일한 수준의 정확도를 유지하며 네트워크를 9x 압축 하였다. 제안한 알고리즘은 비록 원본에 비해 1% 낮은 정확도를 보여주었지만 약 12x의 높은 압축률을 보여주며 낮은 트레이드 오프를 달성 하였다.

Ⅴ. 결 론

우리는 가지치기 알고리즘을 사용함으로써 신경망의 계산량과 메모리 사용률을 줄이는 간단한 방법을 제안했다. 제안한 방법은 다수의 가중치 중 중복값이 존재하고 특징 추출에 미미한 영향을 미치는 값이 존재할 것이라는 판단하에 각레이어의 가중치 평균값을 기준으로 가지치기를 진행한 뒤 나머지 희소 네트워크를 재교육함으로써 실험을 진행하였다. LeNet5와 ImageNet의 AlexNet을 대상으로 실험한 결과, AlexNet의 경우 원본에 비해 정확도가 1% 이내로 손실되었지만 컨볼루션 레이어와 풀리 커넥티드 레이어의 가중치를 12배 이상 압축하였다. 이를 통해 실시간 이미지 처리를 위한 계산량과 메모리 용량이 줄어들어 모바일 플렛폼에 쉽게 이식 가능할 것이다.

ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2017R1D1A3B03033808)

References

[1] 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.

[2] S. C. Lim, S.H. Kim, Y.H. Kim, and D.Y. Kim, “Training Network Design Based on Convolution Neural Network for Object Classification in few class problem,” *Journal of the Korea Institute of Information and Communication Engineering*, vol. 21, no. 1, pp. 144-150, Jan. 2017.

[3] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackal, “Backpropagation applied to handwritten zip code recognition,” *International Journal of Neural Computation*, vol. 1, no. 4, pp. 541-551, Dec. 1989.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, Dec. 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 25th Annual International Conference on Advances in Neural Information Processing Systems*, California:CA, pp. 1106-1114, 2012.

[6] Y. Gong, L. Liu, M.Yang, and L. Bourdev. (2014, December). “Compressing deep convolutional networks using vector quantization,” *arXiv preprint* [Online]. pp.1-10. Available: https://arXiv:1412.6115.

[7] V. Vanhoucke, A. Senior, and M. Z Mao, “Improving the speed of neural networks on cpus,” in *Proceedings of Deep Learning and Unsupervised Feature Learning NIPS Workshop*, Vol. 1. pp.1-8, Dec. 2011.

[8] V. Chandrasekhar, J. Lin, Q. Liao, O. Morère, A. Veillard, L. Duan, and T. Poggio, “Compression of Deep Neural Networks for Image Instance Retrieval,” in *Proceedings of Data Compression Conference*, Utah:UT, pp. 300-309, 2017.

[9] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen, “Compressing neural networks with the hashing trick,” in *Proceedings of International Conference on Machine Learning*, Lille:FR, pp. 2285-2294, 2015.

[10] M. Lin, Q. Chen, and S. Yan. (2014, March). “Network in network,” *arXiv preprint* [Online]. pp.1-10. Available: https://arXiv:1312.4400.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of Computer Vision and Pattern Recognition*, Ohio:OH, pp. 1-9, 2014.

[12] S. Srinivas, and R. V. Babu. (2016, August). "Learning Neural Network Architectures using Backpropagation," *arXiv preprint* [Online]. pp.1-14. Available: https://arXiv:1511.0549.

[13] X. Liu, and T. Yatish. (2016, March). "Pruning of Winograd and FFT Based Convolution Algorithm," *CS231n: Convolutional Neural Networks for Visual Recognition* [Online]. pp.1-7. Available: http://cs231n.stanford.edu/reports/2016/pdfs/117_Report.pdf

[14] Y. Jia, E. Shelhamer, J. Deonahue, S. Karayev, J. Long, R. Girshick, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding." in *Proceedings of the 22nd ACM international conference on Multimedia*, New York: NY, pp. 675-678, 2014.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278 - 2324, Nov. 1998.

[16] E. L Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, “Exploiting linear structure within convolutional networks for efficient evaluation,” in *Proceedings of the 27th Annual International Conference on Advances in Neural Information Processing Systems*, Montreal:CA, pp. 1269- 1277, 2014.

[17] Z. Yang, M. Moczulski, M. Denil, N. de Freitas, A. Smola, L. Song, and Z. Wang, “Deep fried convnets,” in *Proceedings of the IEEE International Conference on Computer Vision*, Massachusetts:MA, pp. 1476-1483, 2015.

[18] S. Han, J. Pool, J. Tran, and W. J. Dally, “Learning both weights and connections for efficient neural networks,” in *Proceedings of the 28th Annual International Conference on Advances in Neural Information Processing Systems*, Montreal:CA, pp. 1135-1143, 2015.

[19] S. Srinivas, and R V. Babu. (2015, July). “Data-free parameter pruning for deep neural networks,” *arXiv preprint* [Online]. pp.1-12. Available: https://arXiv:1507.06149.

[20] M. D Collins, and P. Kohli. (2014. December). “Memory bounded deep convolutional networks,” *arXiv preprint* [Online]. pp.1-10. Available: https://arXiv:1412.1442.

임수창(Su-Chang Lim)

2015년 순천대학교 컴퓨터공학과 졸업(공학사)
2017년 순천대학교 대학원 컴퓨터공학과 졸업(공학석사)
2017년 ~ 현재 순천대학교 대학원 컴퓨터공학과 박사과정
※관심분야 : 컴퓨터비전, 딥러닝, 기계학습

김도연(Do-Yeon Kim)

1986년 충남대학교 계산통계학과 졸업(이학사)
2000년 충남대학교 대학원 정보통신공학과 졸업(공학석사)
2003년 충남대학교 대학원 컴퓨터공학과 졸업(공학박사)
1986년 ~ 1996 한국원자력연구원 선임연구원
1997년 ~ 2008 한국전력기술(주) 책임연구원
2008년 ~ 현재 순천대학교 컴퓨터공학과 교수
※관심분야 : 컴퓨터비전, 딥러닝, 기계학습, 컴퓨터보안