

진화연산 기반 CNN 필터 축소

Evolutionary Computation Based CNN Filter Reduction

서 기 성*
(Kisung Seo)

Abstract - A convolutional neural network (CNN), which is one of the deep learning models, has been very successful in a variety of computer vision tasks. Filters of a CNN are automatically generated, however, they can be further optimized since there exist the possibility of existing redundant and less important features. Therefore, the aim of this paper is a filter reduction to accelerate and compress CNN models. Evolutionary algorithms is adopted to remove the unnecessary filters in order to minimize the parameters of CNN networks while maintaining a good performance of classification. We demonstrate the proposed filter reduction methods performing experiments on CIFAR10 data based on the classification performance. The comparison for three approaches is analysed and the outlook for the potential next steps is suggested.

Key Words : Convolutional neural network, Filter reduction, Genetic algorithm

1. 서 론

딥러닝 기법중의 하나인 CNN (Convolutional Neural Networks)은 생물체의 시각 처리 방식을 응용한 것으로 영상 인식분야에서 탁월한 성능을 보이고 있다[1,2]. CNN은 특징점을 사전에 추출하지 않고, 입력 층에 영상 데이터를 직접 입력하여 컨볼루션 층을 통해 특징이 자동 추출되는 기능을 가지고 있다 [3]. 구체적으로, 다층으로 구성된 다중 필터 기반의 자동적 필터 생성 방식으로 인해, 단일(소수) 필터 및 미분이나 알고리즘을 사용해 필터를 설계하는 전통적인 접근법 [4]과 대비된다.

CNN 기반의 다중 및 다중 필터를 학습에 의해서 개선하면서 물체를 인식하는 것은 이미 뛰어난 성능을 인정받고 있으며 층이 깊어질수록 정확도는 더욱 높아지고 있다[5,6]. GPU등의 딥러닝 연산 환경의 발달로 층과 필터수를 늘린 방대한 규모의 모델이 성능 향상을 주도하고 있으며[7,8], 층의 규모가 커질수록 구조의 최적화가 더 어렵고 방대한 연산량을 필요로 한다. 기존의 경험적인 층 구성에서 벗어나 진화연산을 이용하여 CNN의 구조를 최적화하려는 시도가 연구되어 왔다[9-12]. 더욱 최근인 2018년에는 RNN을 사용하여 CNN 모델을 생성하고 강화학습을 통해 네트워크의 성능을 최대화 하도록 RNN을 학습시키는 NASNet[13] 및 PNASNet[14] 연구가 제안되었고 이전보다 우수한 성능을 보여주고 있다. RNN을 사용한 이유는 CNN의 구조와 연결이 가변길이 스트링으로 적합하게 표현될 수 있기 때문이다. 표현된 CNN을 검증 데이터에 대해서 성능을 평가하고, 이

결과를 보상으로 사용하여 정책 경사(policy gradient) 를 계산하고 RNN을 갱신하는 방식이다[13]. 한편 진화연산을 사용하여 CNN의 구조를 최적화하는 AmoebaNet[15] 연구가 제시되었으며, 방대한 규모의 탐색을 효율화하기 위해서 개체 선택에 나이 개념을 도입하였다(aging evolution by tournament selection). 이 연구는 구글 팀에서 GPU 500대 이상을 10일 이상 투입한 것으로, NASNet/PNASNet과 경쟁할 만한 결과를 얻고 있다[15].

한편으로 CNN 구조의 일부인 필터 수를 줄이는 연구가 시도되고 있다. 이것은 이미 설계된 네트워크에서 컨볼루션 층의 필터중에서 영향력이 적은 것들을 삭제함으로써 전체 네트워크가 차지하는 메모리 용량을 줄이고, 처리 속도를 감소시키는 특성이 있으며, 압축(compression)이나 가지치기(pruning)의 연구로 분류된다[16-21].

CNN 층에서 필터의 삭제를 통해 네트워크의 용량을 축소하는 것은 CNN의 층 구조가 결정된 상태에서 세부적인 개선을 진행하는 것으로, 준 구조적인 최적화에 해당한다. 상기한 대부분의 진화연산 관련 구조 연구가 CNN 층의 설계에 초점을 맞추고 있는 것에 비해, 본 논문에서는 사전 연구[22]에 이어서 진화연산을 기반의 필터 삭제 최적화 알고리즘 연구를 확장하여 수행한다. 필터 삭제에 진화연산을 사용한 연구는 본 연구자 외에는 아직까지 시도되지 않고 있다.

제안하는 진화연산 기반의 필터 축소 접근법은 단지 가중치 기반의 삭제를 통한 네트워크의 축소가 아니라, 필터의 위치 및 전후 관계를 고려하여, 특징에 대한 표현력이 더 향상된 필터 집합을 제공하는데 있다. 즉, CNN이 자동적으로 선택한 필터를 진화연산(GA)를 통해서 정제하여, 적은 수의 필터로서 최대한 성능을 유지하도록 도모한다. 진화연산의 구현시 CNN의 삭제할 필터를 유전자로 표현한다.

6개층(conv 3, 풀링3)으로 구성된 비교적 소규모의 CNN 네트

* Corresponding Author : Department of Electronics Engineering, Seokyeong University, Korea
E-mail: ksseo@skuniv.ac.kr

Received : November 1, 2018; Accepted : November 29, 2018

워크에 대해서 해당 데이터를 충분히 학습시킨 후, 학습된 네트워크의 각층의 필터를 진화연산을 통해 재구성하고 이를 테스트 데이터에 대해서 적합도를 계산하면서 최적화를 진행한다. 필터의 축소율은 전체 필터의 10-30%를 설정하고 일정 세대의 진화연산을 수행하여 분류성능인 정확도(accuracy)의 개선을 비교한다. 부가적으로 최종세대에서 얻어진 축소되고 정제된 CNN 필터들에 대해서 추가적인 학습을 진행하여 성능개선 여부를 관찰한다.

제안된 방법을 CIFAR10에 대해서 실험을 수행하고 원본 네트워크에 대비 분류성능의 유지 수준을 통해 필터 삭제의 효율성을 입증한다.

2. CNN 네트워크 최적화 문제 분석

CNN 네트워크는 그림 1과 같이 특징 추출과 강인화에 관여하는 컨볼루션(convolution) 층과 풀링(pooling) 층이 순차적으로 반복되며, 후반부에 분류 목적에 사용되는 완전연결(fully connected) 층이 위치한다. 일반적인 네트워크 최적화 문제는 구조와 파라미터로 변수를 포함한다. CNN 네트워크에서 CNN 구조에 관한 변수들은 컨볼루션 층수, 풀링 층수, 컨볼루션 층내의 필터 수 및 크기등이 있으며, 패딩, 보폭, 풀링 연산의 방식등이 함께 존재한다. 또한 컨볼루션 층과 풀링 층의 배열에 대한 조합과 층간의 분기도 가능하여 구조의 가능 복잡도는 더욱 증가한다. 네트워크 구조외에 학습률, 가중치 갱신법, 가중치 초기화, 배치 정규화, 활성화 함수등의 하이퍼 파라미터가 부가적으로 존재하나, 이들 중 상당수가 기존 연구에서 상당히 탐색이 된 경우가 많으므로, CNN 네트워크 최적화는 구조적인 문제가 중점으로 볼 수 있다.[9-12, 23]

그런데, 지금까지 기존 구조의 분석과 개선을 통해 많이 알려진 네트워크들이 존재하고, 이들 네트워크들은 일정한 구성 형태를 가지고 있기 때문에, 전체 변화 공간에 대한 임의의 탐색보다는 계산 비용을 고려한 영향력 있는 부분 탐색이 더 효율적일 수 있다. CNN 필터는 정해진 개수만큼 임의로 생성되기 때문에 의미가 떨어지는 필터가 포함될 수 있다. 학습을 통해 필터가 개선되지만, 불필요한 여분이 여전히 존재할 수 있다.

정해진 층 구조에서 층 내의 유용한 필터 선택이나 불필요한 필터를 효율적으로 제거하는 문제는 준구조 최적화에 해당하며, 현실적인 대응 방안이다. CNN 구조/최적화 문제에 대한 분류 개념도가 그림 2에 나와 있다.

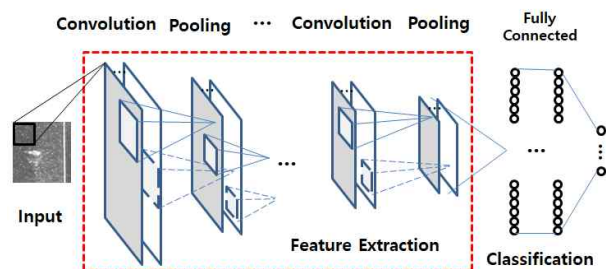


그림 1 CNN 모델 구조
Fig. 1 CNN model structure

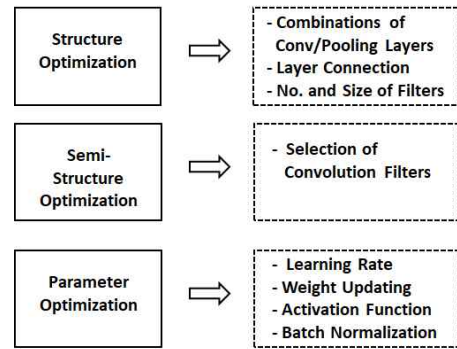


그림 2 CNN 구조/최적화 문제 분류
Fig. 2 Classification of CNN structure/parameter optimization

3. CNN 필터 선택 최적화

대부분의 구조 또는 파라미터 최적화 문제는 방대한 경우의 수로 구성된 탐색 공간 때문에 진화연산 방식이 사용되고 있다. 또한 문제의 표현에 있어서 CNN 필터의 구성과 유전자 표현이 부합되므로, 본 연구에서는 진화연산을 사용한 필터 최적화 문제를 다루고자 한다. 특히, DNN의 각 노드가 하나의 가중치 값으로 연결된 것에 비해서 CNN의 필터는 $n \times n$ 개의 가중치로 구성되어 있기 때문에 하나의 필터라도 이를 단순한 가중치 값을 기준으로 처리하기 힘들다.

3.1 진화 연산

진화연산은 유전자 표현, 유전자 생성, 적합도 평가, 선택, 유전연산 부분으로 구성되어 있다. 이 중에서 구조/파라미터 최적화 문제에 대해서 밀접하게 관련된 부분은 해의 유전자 표현 방법과 생성 및 탐색이다. 본 논문에서 다루는 필터의 선택은 조합 최적화 문제에 해당하며, GA (Genetic Algorithm) 방식으로 구현한다. 선택 또는 삭제할 필터들을 GA 스트링으로 표현한다.

GA의 수행과정은 다음과 같다. 초기에 스트링으로 구성된 개체들을 임의로 생성한다. 그리고 각 개체를 해석하여 구한 후보 해를 적합도 함수로 평가한다. 이후, 주어진 선택 방법에 의해 유전 연산에 참여할 개체들을 선택한다. 선택된 개체들을 대상으로 유전 연산(교배, 돌연변이) 수행한다. 그리고 이 전체과정을 종료조건이 만족될 때까지 반복한다. GA에서의 교배 연산자는 임의의 점에서 부모의 스트링의 일부를 교체하는 방식으로 수행된다.

3.2 필터의 선택/삭제 문제의 유전자 표현

컨볼루션 층에서 필터의 선택과 삭제는 동전의 양면이다. 필터의 표현(representation) 측면에서 보면 좋은 필터를 선택하는 것이 CNN 네트워크의 성능을 향상시킬 수 있는데, 이것은 필요성이 떨어지는 필터를 삭제하고 남는 상태와 같다. 압축(compression)이나 가지치기(pruning)의 관점에서는 삭제 작용이

더 부합된다. 본 논문에서는 필터의 선택 및 삭제의 의미가 서로 상대적이므로 경우에 따라 혼용하기로 한다. 상대적으로 불필요한 필터를 삭제하기 위해서 각 층의 필터 수 범위 내에서 필터 번호를 유전자로 구성한다.

그림 3과 같이 n 개의 층으로 구성된 CNN 네트워크의 각 컨볼루션 층마다 삭제할 필터 번호로 구성된 스트링이 대응된다. 유전형은 삭제할 번호를, 반대로 표현형은 삭제되고 남은 필터로 구성되어 성능을 평가한다. 선택할 필터보다 삭제할 필터 수가 적기 때문에 유전자 길이를 줄이기 위해서 삭제할 필터를 유전자로 표현한다. 그림 3에서 유전자 맨 앞의 숫자 2는 conv 1층에서 2번째의 필터를 삭제한다는 의미이다. 회색 음영은 삭제할 필터를, 흰색은 보존할 필터 번호를 나타낸다.

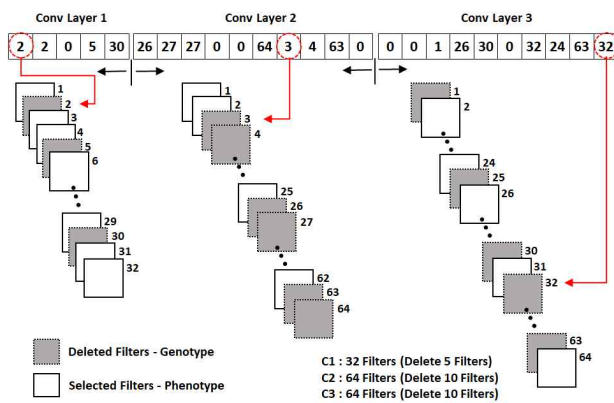


그림 3 CNN 필터 삭제를 위한 유전자 표현
Fig. 3 GA chromosome for deleting CNN filters

3.3 유전자 공간 탐색

그림 3은 단지 3개의 컨볼루션 층으로 구성된 소규모의 CNN 네트워크를 예로 들었지만, 컨볼루션 층이 늘어나거나 층 당 필터수가 증가된 대규모의 네트워크일 경우, 유전자의 길이는 매우 커진다. 이에 따라 탐색 공간이 방대해지며, 이로 인하여 교배나 변이 연산에서 탐색 공간의 이동이 효율적이지 못하다. 이를 보완하기 위해서 일반적인 1점 교배 연산자 외에 다양한 탐색을 위해서 2점 교배 연산자를 함께 사용한다(그림 4). 또한, 돌연변이

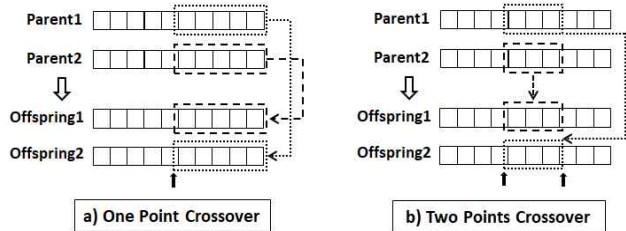


그림 4 유전 연산자 a) 1점 교배, b) 2점 교배
Fig. 4 Genetic operations a) one-point crossover, b) two-point crossover

연산자의 경우도 한 개체 내에서 다수의 요소가 변이 될 수 있도록 구현한다.

3.4 진화연산 수행 방식

진화연산을 사용한 CNN 구조 최적화에 대한 구글 팀등의 일부 연구에서는 500여대의 대규모 GPU를 사용하여 각 개체마다 학습과 테스트를 수행하였지만, 본 논문에서는 제한된 컴퓨팅 자원으로 효율적인 진화연산을 수행하기 위해서, 학습 없이 테스트만으로 개체의 적합도를 평가한다.

Algorithm 1 Evolution of CNN Filter Deletion

- 1: CNN training for train dataset
- 2: $t \leftarrow 0$
- 3: initialize $P(t)$ // individuals for filter selection
- 4: **procedure** Evaluate $P(t)$
- 5: CNN test for test dataset
- 6: calculate accuracy
- 7: **while** not termination-condition **do**
- 8: $t \leftarrow t+1$
- 9: select $P(t)$ from $P(t-1)$
- 10: crossover and mutation $P(t)$
- 11: **procedure** evaluate $P(t)$
- 12: CNN test for test dataset
- 13: calculate accuracy

진화연산 기반의 CNN 필터 삭제를 위한 전체 알고리즘은 Algorithm 1과 같다. 사전에 학습 데이터에 대해서 학습된 CNN 네트워크를 대상으로 삭제할 필터 유전자를 개체로 생성하고, 테스트 데이터에 대해서 이들의 정확도 성능을 평가한다. 진화 연산의 세대를 증가시키면 개체를 선택하고 교배와 돌연변이 연산을 수행하며, 삭제 후 CNN의 성능 평가 과정을 반복적으로 수행한다.

본 논문에서는 다음과 같이 3가지 진화연산 수행 방식을 제안하고 이를 비교한다(그림 5). 첫째, 사전에 정해진 삭제율(15~30%) 만큼의 필터를 한 번에 삭제하고, 이후 충분한 세대(500 epochs)를 재학습시켜 필터 삭제에 따라 떨어진 분류 정확도 성능을 끌어 올리도록 한다(그림 5 상단). 둘째, 각 컨볼루션 층에 대해서 순차적으로 삭제하는데, 첫 층만을 삭제하고 일부 재훈련(100 epochs)을 수행한 후 두 번째 층을 삭제한다. 다시 재훈련을 수행 후, 이를 다음 층에 대해서 반복한다(그림 5 중단). 셋째, 정해진 삭제율을 n 번으로 나누어 수행하고 그 사이마다 재훈련(100 epochs)을 수행한다.

사전 실험을 통해 이와 같은 방식의 변화가 성능에 상당한 영향을 미칠 수 있음을 파악하였고, 이를 개선하여 방법화하고 제안한다.

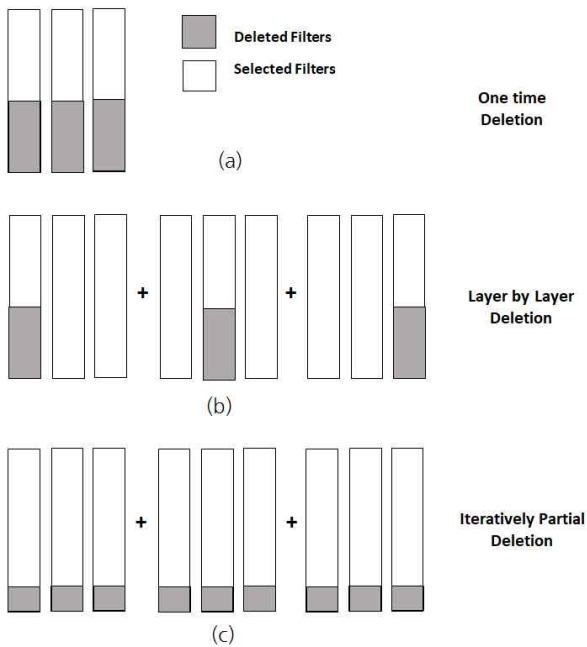


그림 5 필터 삭제 3 가지 방법 (a) 전체 한번 삭제, (b) 층별 삭제, (c) 반복적 부분 삭제

Fig. 5 Filter deletion methods (a) one time deletion (b) layer by layer deletion (c) iteratively partial deletion

4. 실험

CNN 필터 삭제 실험은 표 1에 나온 Conv12 네트워크에 대해서 CIFAR10 데이터를 사용한다. 진화 연산 과정은 다음과 같은 GA 파라미터를 사용하여 GTX-1080Ti GPU 환경에서 수행된다.

표 1 Conv12 네트워크의 파라미터

Table 1 Parameters of Conv12 model

Layer	Output Shape	Filter Size	Parameters
Input	32 x 32 x 3		0
Conv1	31 x 31 x 32	2x2	0.42K
Pool1	15 x 15 x 32	2x2	0
Conv2	14 x 14 x 64	2x2	8.26K
Conv2	13 x 13 x 64	2x2	16.45K
Pool2	6 x 6 x 64	2x2	0
FC			295.04K
Output			1.29K

실험에 사용된 GA 파라미터는 다음과 같다.

1. Number of generations: 20
2. Population sizes: 50
3. Selection: Tournament (size=7)
4. Crossover: 0.85
5. Mutation: 0.2

표 2는 GA 기반의 전체 한번 삭제 방법으로 전체 필터의 15.6%를 삭제한 후 진화연산 최종세대에서 원본 네트워크의 89.7% 정확도를 얻었고, 삭제 후 네트워크를 500 epochs 재학습하여 98.3%의 성능을 얻었다. 이를 통해 본 연구에서 시도한 GA 기반의 필터 삭제 방법이 유효함을 알 수 있다.

표 2 GA 기반 필터 축소-전체 한번 삭제(15.6% 삭제율)

Table 2 GA-based filtering reduction 1-one time deletion (15% reduction rate)

ACC Data	Original	GA Filter Selection	Retrain after GA Filter Selection (epoch 500)
CIFAR-10	64.60%	57.94% (89.7%)	63.55%(98.3%)

표 3은 같은 방법으로 전체 필터의 31.3%를 삭제한 후 원본 네트워크의 83.2% 정확도를 얻었고, 삭제후 네트워크를 500 epochs 재학습하여 93.4%의 성능을 얻었다. 표 1에 비하여 필터를 두배 가량 삭제했으므로 당연히 분류 성능이 많이 저하되었고, 재학습 후에도 표 1에 비해서 원본 네트워크 성능과 6.4% 차이남을 알 수 있다. 그러나 삭제율에 비해서는 성능 저하가 적으므로 적은 규모의 네트워크와 처리 속도의 감소를 필요로 하는 경우에는 유용하다고 볼 수 있다.

표 3 GA 기반 필터 축소-전체 한번 삭제(15.6% 삭제율)

Table 3 GA-based filtering reduction 2 - one time deletion (31% reduction rate)

ACC Data	Original	GA Filter Selection	Retrain after GA Filter Selection (epoch 500)
CIFAR-10	64.60%	53.74% (83.2%)	60.33%(93.4%)

제안된 3가지 진화연산 방법의 비교 결과가 표 4에 나와 있다. 전체 삭제(One time deletion) 방법에 비해 층별 삭제(Layer by layer)와 반복적 부분 삭제(Iteratively Partial Deletion)가 훨씬 더 우수하였으며, 15%와 31% 삭제율 간의 성능 차이도 매우 적었다. 이는 일부 삭제와 재훈련을 통해서 지속적인 최적화가 수행되었기 때문으로 보인다. 반복적 부분 삭제가 층별 삭제에 비해서도 약간 더 우수한 성능을 나타내었다.

표 4 3가지 GA 기반 필터 축소 방법의 비교(15%/31% 삭제율)

Table 4 Comparison of three GA-based filter reduction methods (15%/31% reduction rate)

ACC Reduction Rate	One Time Deletion	Layer by Layer Deletion	Iteratively Partial Deletion
15%	89.7%	99.6%	99.8%
15% (Retrain)	98.3%	99.7%	99.9%
31%	83.2%	97.7%	98.6%
31% (Retrain)	93.4%	98.7%	99.2%

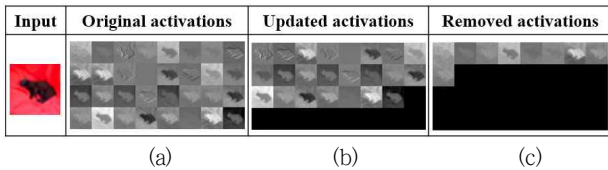


그림 6 활성화 맵의 시각화 (a) 원본 맵, (b) 삭제후 맵, (c) 제거된 맵

Fig. 6 Visualization of the activation map (a) original activations (b) updated activations (c) removed activations

CIFAR-10 데이터 중 고양이 이미지에 대한 활성화 맵이 그림 6에 시각화되어 있다. 좌측에 원본 맵이 나와 있고, 중간에 진화연산 기반의 반복적 부분 삭제를 통해 얻은 필터에 해당하는 활성화 맵이 나와 있다. 아래 검은색 부분은 삭제된 필터에 대응하는 것이라 검게 표시되어 있다. 원본 맵에 비해서, 활성화 정도가 약한 맵들이 제거되고, 대부분 유의미한 맵들로 구성되어 있음을 알 수 있다. 우측은 제거된 필터에 해당하는 활성화 맵이다.

6. 결 론

CNN 네트워크의 성능을 유지하면서, 용량과 계산비용을 축소하기 위해서 진화연산 기반의 필터 삭제 방법을 제안하였다. 진화적 필터 삭제와 재훈련 과정을 증별 수행과 반복적 부분 삭제 방식을 구현하고 전체 한번 삭제 방식과 비교하였다. 6개층(conv 3, 풀링3)으로 구성된 소규모의 CNN 네트워크를 사용하여 CIFAR10 데이터에 대해서 실험을 수행하였고, 원본 네트워크에 대비 분류 성능은 99%, 축소율은 최대 31%를 얻을 수 있었다. 이를 통해 진화연산을 필터 삭제에 적용할 수 있는 가능성을 발견하였고, 향후 축소율의 확대 및 가중치 기반의 가지치기(pruning) 알고리즘과의 비교도 수행할 예정이다.

감사의 글

본 연구는 2017학년도 서경대학교 교내연구비 지원에 의하여 이루어졌음.

References

- [1] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview", *Neural Networks*, Vol. 61, pp. 85-117, 2015.
- [2] Y. LeCun, Y. Bengio, G. Hinton, "Deep learning," *Nature*, Vol. 521, pp. 436-444, 2015.
- [3] LeCun, Yann, et al. "Gradient based learning applied to document recognition", *Proceedings of the IEEE*, pp. 2278-2324, 1998.
- [4] Yunwon Park, In So Kweon "Ambiguous Surface Defect Image Classification of AMOLEDD is playsin Smartphones", *IEEE Trans. Industrial Informatics*, 12(2): 597-607,2016
- [5] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *International Conference on Learning Representations*, 2014.
- [6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions", *Computer Vision and Pattern Recognition*, 2015
- [7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", *Computer Vision and Pattern Recognition*, 2016..
- [8] S. Zagoruyko and N. Komodakis, "Wide Residual Networks", arXiv: 1605.07146, 2016.
- [9] C. Fernando et al. "Convolution by Evolution: Differentiable Pattern Producing Networks", *In Proceedings of the 2016 Genetic and Evolutionary Computation Conference*, Denver, CO, USA, pp. 109-116. 2016.
- [10] A. Rikhtegar, M. Pooyan, M. Manzuri-Shalmani, "Genetic algorithm-optimised structure of convolutional neural network for face recognition applications", *IET Computer Vision*, Vol. 10, Iss. 6, pp. 559-566, 2016
- [11] L. Xie, A. Yuille, "Genetic CNN", *CVPR 2017*
- [12] M. Sukanuma, s, Shirakawa, T. Nagao, "A Genetic Programming Approach to Designing Convolutional Neural Network Architectures", *Proceedings of GECCO 2017*, pp. 497-504, 2017
- [13] B. Zoph, and Q. V. Le, "Neural Architecture Search with Reinforcement Learning", *CoRR abs/1611.01578* 2016
- [14] C. Liu, B. Zoph, J. Shlens, W. Hua, L. J. Li, L. Fei-Fei, and K. Murphy, "Progressive Neural Architecture Search", *ECCV 2018*
- [15] E. Real, A. Aggarwal, Y. Huang, Q. V. Le, "Aging Evolution for Image Classifier Architecture Search", 2018
- [16] H. Hu, R. Peng, Y.W. Tai, and C.K. Tang. "Network trimming: A data-driven neuron pruning approach towards efficient deep architectures", arXiv preprint arXiv:1607.03250, 2016.
- [17] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning Filters for Efficient ConvNets", *CoRR abs/1608.08710* (2016).
- [18] Q. Huang, K. Zhou, S. You, and U. Neumann, "Learning to prune filters in convolutional neural networks", arXiv preprint arXiv:1801.07365. 2018 Jan 23.
- [19] C. Chen, F. Tung, N. Vedula, and G. Mori, "Constraint-Aware Deep Neural Network Compression", *ECCV (8)* 2018: 409-424.
- [20] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding", arXiv preprint arXiv:1510.00149, 2015.
- [21]] J. H. Luo, J. Wu, and W. Lin, "ThiNet: A Filter Level

Pruning Method for Deep Neural Network Compression”, *ICCV 2017*: 5068-5076.

- [22] J. Kim, M. Lee, J. Choi, K. Seo, “GA-based Filter Selection for Representation in Convolutional Neural Networks”, *ECCV 2018 Workshop on Compact and Efficient Feature Representation and Learning in Computer Vision*.
- [23] K. Seo, “Analysis of evolutionary optimization methods for CNN structures”, *Transactions of the Korean Institute of Electrical Engineers*, 67(6), pp. 767-772, 2018

저 자 소 개



서 기 성 (Kisung Seo)

1986년 연세대학교 전기공학과 졸업(학사).
1988년 연세대학교 대학원 전기공학과 졸업(석사). 1993년 연세대학교 대학원 전기공학과 졸업(박사). 1999~2003년 Michigan State University, Genetic Algorithms Research and Applications Group, Research Associate, 2002~ 2003년 Michigan State University, Electrical & Computer Engineering, Visiting Assistant Professor, 2011~2012년 Michigan State University, BEACON (Bio/ computational Evolution in Action CONSortium) Center, Visiting Scholar, 1993년~현재 서경대학교 전자공학과 교수. 관심분야는 진화연산, 딥러닝, 머신비전, 가상예측, 지능로봇