

Self-Attention 연산을 적용한 Neural Architecture Search

신은섭[○] 배성호

경희대학교

kairos9603@khu.ac.kr, shbae@khu.ac.kr

A New Neural Architecture Search Using Self-Attention Operation

Eunseop Shin[○] Sung-Ho Bae

Kyung Hee University

요 약

NAS(Neural Architecture Search)는 뉴럴 네트워크가 스스로 데이터에 맞는 네트워크를 찾는 Auto-ML 방법의 하나이다. 기존의 NAS는 Convolution, Pooling 등 간단한 연산을 조합하여 네트워크를 찾았다. Attention 모듈은 사람이 손으로 디자인한(Hand-crafted) 모델에서 좋은 성능을 보였으며, 네트워크가 어떤 영역에 집중하여 결정을 내리는지도 알 수 있게 해주는 좋은 모듈이다. 본 논문에서는 Self-Attention 연산을 NAS에 추가하여 Architecture Search를 수행하였다. 실험 결과 97.24%의 정확도로 기존의 DARTS 보다 0.01% 향상된 네트워크 구조를 찾았고, 이로서 Self-Attention 연산이 NAS에서도 좋은 효과를 보임을 증명하였다.

1. 서 론

NAS(Neural Architecture Search)[1, 2, 3]는 Auto-ML 분야 중 뉴럴 네트워크 구조를 자동으로 찾아주는 모델을 연구하는 분야이다. DARTS[3]는 미분 불가능했던 NAS문제를 Continuous Relaxation과 DAG(Directed Acyclic Graph)를 적용하여 미분 가능하게 바꾸었다. 이로 인해 기존의 뉴럴 네트워크를 학습하는 것과 동일한 방법인 Gradient Descent 방법을 사용하여 NAS를 학습할 수 있게 하였다. 이로 인해 강화 학습 방법으로 NAS를 학습하던 기존의 방법으로는 수 일이 걸리던 학습 시간을 수시간 내로 줄일 수 있게 되었다.

NAS 이전에 사람이 직접 모델을 설계하였고(Hand-Crafted) 그 성능을 높이기 위한 다양한 시도가 있었다. 그 중 가장 주목받는 연산인 Self-Attention은 SENet[4]에서 본격적으로 사용되어 ILSVRC 2017에서 1등이라는 성과를 냈다.

기존의 NAS는 3x3 Convolution, Max Pooling, Average Pooling, Skip-Connection 등 간단한 연산을 이용하여 네트워크 구조를 찾았다. 이에 본 논문은 NAS에 Self-Attention 연산인 Channel-Attention과 Spatial-Attention을 추가하여 좋은 성능의 네트워크를

탐색하고, 탐색한 네트워크의 구조를 분석하는 연구를 제안한다.

1. 기존 연구

1.1 DARTS

기존의 NAS는 Objective Function의 미분이 불가능하였다. 따라서, 강화학습이나 유전 알고리즘 등의 방법으로 학습을 진행하였다. 그러나 이 방법들은 리소스가 매우 많이 필요하고 학습시간도 오래 걸리는 문제점이 있었다. DARTS[3]는 이런 문제를 해결하기 위해 미분이 불가능했던 Objective Function을 Continuous Relaxation 방법을 이용하여 미분 가능하게 만들었다. Continuous Relaxation 방법은 기존의 여러 연산 중 하나를 선택하기 위해 MAX 값을 사용했던 것을 Softmax 값으로 대체한 방법이다. 이 방법으로 Discrete하게 결정되었던 연산들이 Continuous 해지면서 Objective Function이 미분 가능한 형태가 되었다. 이를 이용해 Super Network를 DAG(Directed Acyclic Graph) 형태로 구현하였다. 그래프의 Edge가 Operation을 Weight를 나타내고, 학습 중에는 모든 Edge의 weight를 학습하지만 학습이 끝난 후에는 가장 큰 weight를 가진 하나의 Edge만이 선택된다.

Objective Function이 미분가능하기 때문에 기존의 딥러닝 학습에서 사용하였던 Gradient Descent 방법을 이용하여 Super Network를 학습할 수 있게 되었고, 기존의 방법들보다 더 빠르고 높은 성능을 보이는 네트워크를 생성할 수 있었다.

* "본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학 사업의 연구결과로 수행되었음"(2017-0-00093)

또한, DARTS[3]는 빠른 학습을 위해 NasNet Search Space[2]를 사용한다. NasNet Search Space[2]는 전체 네트워크를 실행하는 것이 아닌 네트워크에서 반복적으로 사용될 수 있는 Cell을 찾는다. Cell은 크게

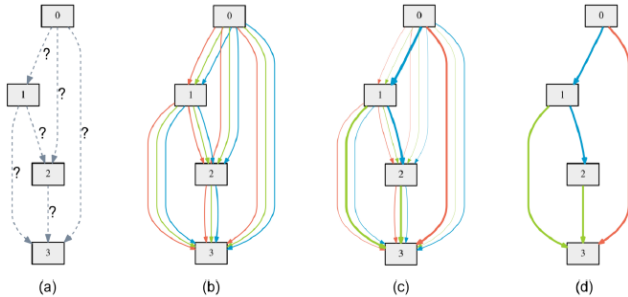


그림 1. DARTS[3]에서 사용되는 Super Network

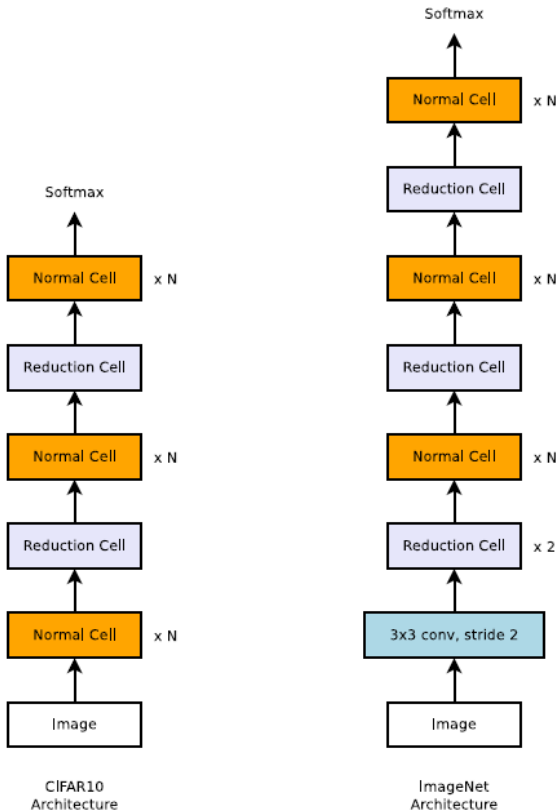


그림 2. NasNet Space Architecture

Normal cell과 Reduction cell로 나뉘며, Reduction cell에서만 출력 feature의 spatial size가 반으로 줄어든다. 전체 네트워크는 그림 2와 같이 Normal cell과 Reduction cell을 번갈아 가면서 실행한다. 이렇게 찾아진 Cell은 Image Net과 같이 크기가 큰 데이터셋을 학습하기 위한 네트워크를 구성할 때에도 다시 Cell을 찾는 과정없이 여러 번 반복하여 사용이 가능하다.

1.2 Self-Attention

Attention은 NLP[5]에서 많이 사용되는 개념으로 긴

문맥에서 사람이 집중적으로 보는 부분이 있고 그 부분이 맥락을 파악하는데 중요한 역할을 한다는 점에서 아이디어를 얻어 만들어낸 구조이다. 컴퓨터 비전 분야에서도 이 아이디어를 이용하였다. 대표적으로 사용된 곳이 SENet[4]과 BAM[5], CBAM[6]이다.

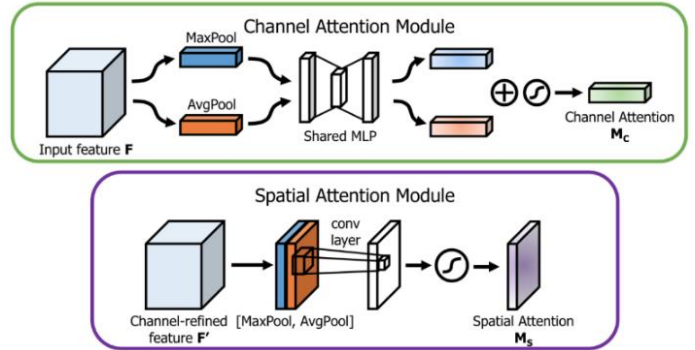


그림 3. CBAM에서 제안하는 두 가지 Attention Module

특히 BAM[5]과 CBAM[6]에서는 Channel Attention과 Spatial Attention을 결합하여 모델에서 다음스테이지로 넘어갈 때 더 정확한 정보를 전달, 성능을 향상시켰다.

2. 제안하는 모델

2.1 기존 연구와 차이점 및 해결방안

DARTS[3]의 경우 DAG를 구성하는 연산 중에 Attention이 없다. BAM[5]과 CBAM[6]을 통해서 Attention이 기존의 뉴럴 네트워크에서 효과적인 역할을 하는 것을 볼 수 있었다. 본 연구에서는 기존 DARTS[3]의 연산에 두가지 Attention모듈을 추가하고 학습하여 Attention 모듈이 선택되었는지, 선택되었다면 어떤 Attention 모듈이 선택되었는지를 확인하고 최종적으로 찾아진 네트워크의 성능을 분석한다.

2.2 Attention Module

DARTS[3]에 CBAM[7]에서 제안한 다음 두가지 Attention 모듈을 추가하여 학습을 시켜본다. 첫번째 모듈은 Spatial Attention 모듈로 Spatial 영역에서 Attention 해야 하는 위치를 찾는 모듈이다. 입력의 모든 채널을 1로 Pooling하여 그 값을 원본 입력과 곱하여 Attention을 계산한다. 이를 통해 Spatial 영역에서 중요한 부분은 더욱 부각되고 중요하지 않은 부분은 무시되는 효과를 얻을 수 있다. 두번째로 Channel Attention은 모든 Spatial 영역을 1x1로 Pooling 한 뒤 Spatial Attention과 마찬가지로 원본과 곱하여 Attention을 계산한다. Convolution 레이어를 통해서 학습된 Channel에 대해서 Attention을 하며, 더 좋은 Feature Channel에 대해서 집중하도록 하는 효과가 있다.

