물체 탐지에서 Neural Architecture Search 기반 Channel Pruning 을 통한 Parameter 수 대비 정확도 개선

노재현[1], 유승현[2], 손승욱[3], 정용화[4]
[1]고려대학교 컴퓨터융합소프트웨어학과 학부생
[2]고려대학교 컴퓨터융합소프트웨어학과 석사과정
[3]인포벨리코리아 기업부설 인공지능 세종연구소 사원
[4]고려대학교 컴퓨터융합소프트웨어학과 교수

tngkrshsh@korea.ac.kr[1], tidlsld44@korea.ac.kr[2], sso7199@invako.kr[3], ychungy@korea.ac.kr[4]

Improving Accuracy over Parameter through Channel Pruning based on Neural Architecture Search in Object Detection

Jaehyeon Roh*, Seunghyun Yu*, Seungwook Son**, Yongwha Chung*
*Dept of Computer Convergence Software, Korea University
**INFO VALLEY KOREA

요 약

CNN 기반 Deep Learning 분야에서 객체 탐지 정확도를 높이기 위해 모델의 많은 Parameter 가 사용된다. 많은 Parameter 를 사용하게 되면 최소 하드웨어 성능 요구치가 상승하고 처리속도도 감소한다는 문제가 있어, 최소한의 정확도 하락으로 Parameter 를 줄이기 위한 여러 Pruning 기법이 사용된다. 본 연구에서는 Neural Architecture Search(NAS) 기반 Channel Pruning 인 Artificial Bee Colony(ABC) 알고리즘을 사용하였고, 기존 NAS 기반 Channel Pruning 논문들이 Classification Task 에서만 실험한 것과 달리 Object Detection Task 에서도 NAS 기반 Channel Pruning 을 적용하여 기존 Uniform Pruning 과 비교할 때 파라미터 수 대비 정확도가 개선됨을 확인하였다.

1. 서론

Channel Pruning 은 Deep Learning 모델의 경량화 방법 중의 하나이다. 최신 Channel Pruning 기법은 Regularization, Weight, Activation 등 다양한 기준[1]을 통해 모델의 채널(필터)을 선택하는 것에 중점을 두었으나, 이러한 기준들은 저자들이 가정한 실험을 통해 만들어진 것이다. 이는 Pruning 된 결과가 최적의 결과가 아닐 수도 있다는 것을 의미하며, 이를 개선하기 위해 Neural Architecture Search(NAS) 기법을 사용한 Pruning 연구들이 진행되었다. 그러나, 기존 NAS 기반 Channel Pruning 연구들[1]은 대부분 Classification Task 에서 실험되어, Object Detection Task 에서도 파라미터 수 대비 정확도가 개선되는지를 확인하기 위하여 본 연구에서는 YOLOv7-Tiny[2]에 Artificial Bee Colony (ABC)[3] NAS 알고리즘을 적용한 실험을 수행한다.

2. 관련 연구

Pruning 시 Weight 를 가져오는 방식과 Pruning 방식에 관하여 여러 실험이 발표되었다. 우선, 기존 모델에서 채널을 선택하는 기준을 선정하여 Pruning 하는 것보다 Random Weight 를 사용하여 Pruning 하는 것이 더 높은 정확도를 가진다. 또한, Random Weight 를 사용하되 모든 레이어를 미리 정한 비율로 Uniform 하게 Pruning 하는 것보다 레이어 별로 다르게 Pruning 하는 것이 파라미터 수 대비 정확도가 개선된다[3]. 또한 위 실험결과를 바탕으로 ABC 알고리즘을 사용한 Channel Pruning 실험이 진행되었다[4].

위 연구들뿐만 아니라 기존의 NAS 기법을 적용한 Pruning 연구들은 대부분 Classification Task 에서 실험되어, Object Detection Task 에서도 기존의 Uniform Pruning 방식에 비해 NAS 기법이 파라미터 수 대비 정확도가 개선될 수 있는지 확인이 필요하다.

3. ABC 알고리즘

ABC 알고리즘의 첫번째 단계는 set $C_n = [x_1, \ldots, x_L]$으로 구성되어 있는 set C 를 초기화하는 것이다. x 는 $1 <= x <= U(2<=U<=10)$인 정수이며, 이는 각 레이어의 채널 수가 (기존 채널 수)*(x/10)로 바뀌는 것을 의미한다.

Artificial Bee Colony(ABC) Algorithm

Input: Iterations: I, Upbound: U, Number of Bees: N, Counter: T, Max Trial: M, Total Number of Convolution Layers: L
Output: Optimal pruned structure code C*
fit(Fitness): $0.1*AP_{50} + 0.9*AP_{50:95}$ (YOLOv7 Standard)

```
0  Initialize the pruned structure code set C;
1  for h = 1 → I do
2      for p = 1 → N do
3          generate new code Gp that slightly different from Cp;
4          calculate fitness of Gp and Cp;
5          if fitGp > fitCp then
6              Cp = Gp;
7              fitCp = fitGp;
8              Tp = 0;
9          else
10             Tp = Tp + 1;
11         end
12     end
13     for i = 1 → N do
14         calculate probability Pi;
15         generate a random real number Ri ∈ [0, 1];
16         if Ri <= Pi then
17             generate new code Gi that slightly different from Ci;
18             calculate fitness of Gi;
19             if fitGi > fitCi then
20                 Ci = Gi;
21                 fitCi = fitGi;
22                 Ti = 0;
23             else
24                 Ti = Ti + 1;
25             end
26         end
27     end
28     for j = 1 → N do
29         if Tj > M then
30             re-initialize Cj;
31         end
32     end
33 end
34 return Optimal pruned structure code C*
```

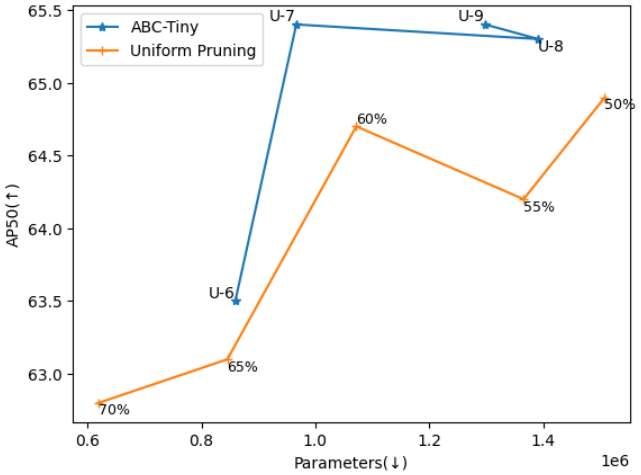
두번째 단계인 Employed Bee 단계(Line 2-12)에서, 각각 1 개의 code set 을 할당 받고 그 set 의 neighbor set 을 탐색한다. 이후, fitness 를 구한 뒤 기존 set 의 fitness 보다 크면 set 을 교체하고 아니라면 삭제한다.

세번째 단계인 Onlooker Bee 단계(Line 13-27)에서, 이전 단계의 결과를 바탕으로 자신이 탐색할 set 을 확률적으로 선택한다. 이후 선택된 set 에 대해 두번째 단계를 적용한다.

네번째 단계인 Scout Bee 단계(Line 28-32)에서, 각각 set 의 trial 을 검사하고 Max Trial M 보다 커지면 set 을 초기화하여 새로운 code set 을 탐색하도록 한다.

4. 실험 결과

본 논문의 실험은 GPU: RTX 3090, Baseline Model: YOLOv7-Tiny, Dataset: Argoverse[5]로 실험하였다. 그림 1 과 표 1 에서 알 수 있듯이 ABC 알고리즘을 적용한 모델들은 Random-Weight Uniform-Pruning 모델들에 비해 파라미터 수 대비 정확도가 개선되었다.



<그림 1> 기존 Random Weight-Uniform Pruning 과 ABC-Tiny 의 파라미터 수 대비 정확도 비교 그래프. 파란 그래프의 U-X 는 Upbound U 값을 X 만큼 설정했다는 의미이고, 주황 그래프의 X%는 Channel Pruning Rate 를 의미한다.

Model	Parameter	Pruned %(↑)	AP_{50}	Reduced %(↓)
Baseline	6.01 M	-	69.4	-
U-8	1.40 M	**76.7**	65.3	**5.9**
Uniform-50%	1.51 M	74.9	64.9	6.5
U-9	1.30 M	**78.4**	65.4	**5.8**
Uniform-55%	1.37 M	77.2	64.2	7.5
U-7	0.97 M	**83.9**	65.4	**5.8**
Uniform-60%	1.07 M	82.2	64.7	6.8
U-6	0.86 M	85.6	63.5	**8.5**
Uniform-65%	0.85 M	**85.9**	63.1	9.1
Uniform-70%	0.62 M	89.7	62.8	9.5

<표 1> 기존 Pruning 과 ABC-Tiny 의 파라미터와 정확도 감소율 비교 표. Pruned 와 Reduced 는 파라미터와 정확도가 각각 Baseline 대비 몇 % 감소하였는지를 의미한다.

감사의 글

이 논문은 2023 년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구입니다. (PP00241770, 2023 년 지역혁신클러스터육성)

참고문헌

[1] He Y., Xiao L., "Structured Pruning for Deep Conv-olutional Neural Networks: A survey," *arXiv*, 2023, arXiv:2303.00566.
[2] Wang C., Bochkovskiy A., Liao H., "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp.7464-7475.
[3] Lin M., Ji R., Zhang Y., Zhang B., Wu Y., Tian Y., "Channel Pruning via Automatic Structure Search," In *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2020, arXiv:2001.08565.
[4] Liu Z., Sun M., Zhou T., Huang G., Darrell T., "Rethinking the value of Network Pruning," In *Proceedings of the International Conference on Learning Representations*, 2019, arXiv:1810.05270.
[5] Argoverse-HD. Available at: https://www.kaggle.com/datasets-/mtlics/argoversehd