2024연세대학교 공학대학원 인공지능 전공 AI SHOWCASE

강화학습 Computer Vision 모델 압축

김장박이정

김병주 2023450017
장지선 2022451109
박병선 2024451022
이지아 2024451036

지도교수: 함범섭 교수님

Introduction

Hardware Issue

- Target Board 메모리 용량이 작은 경우
  대용량의 고도화된 딥러닝 모델을 올릴 수 없음

Cost Issue

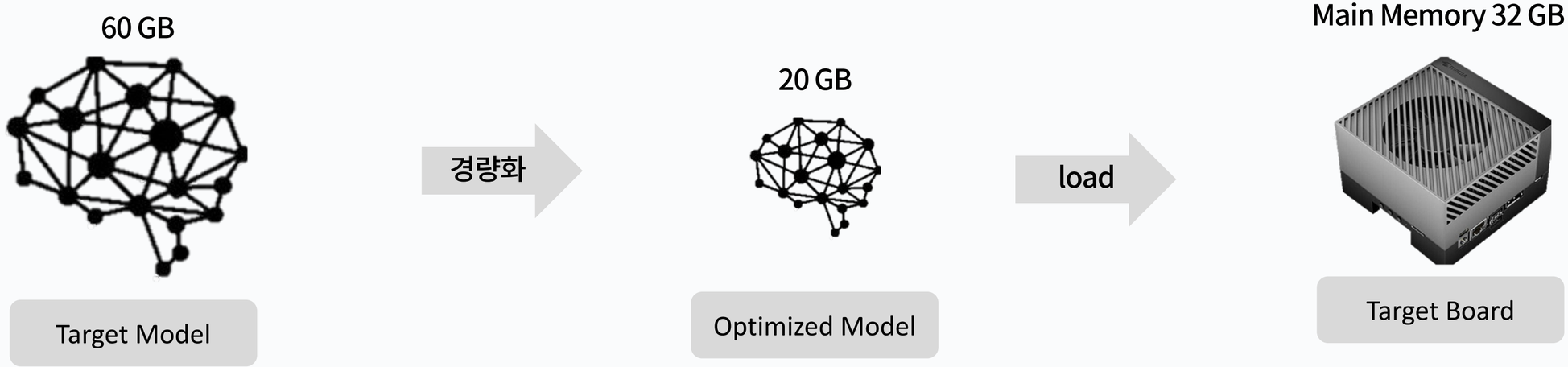
- 고도화된 모델은 파라미터 갯수가 많음
- 상대적으로 비싼 대용량의 메모리 필요

Real-time Issue

- 연산량 감소로 inference time이 짧아짐
- 보다 나은 서비스 제공 가능

Environment Issue

- GPU의 큰 전력소모로 다량의 화석에너지 사용
- 환경 이상기후 발생



Previous work

기존 연구들

학습된 Network 통계를 이용한 pruning

➡ 압축률을 정해놓고 압축하여 새로운 시도가 부족

채널 중요도를 이용해 학습한 pruning

➡ Loss Function이 달라 Optimizer를
기존 네트워크와 다르게 사용해 반복적인 fine-tuning 필요

Architecture Search를 이용한 pruning

➡ 일반적인 딥러닝에 비해 연산 cost가 높음

VS.

DECORE 적용된 PRUNING 장점

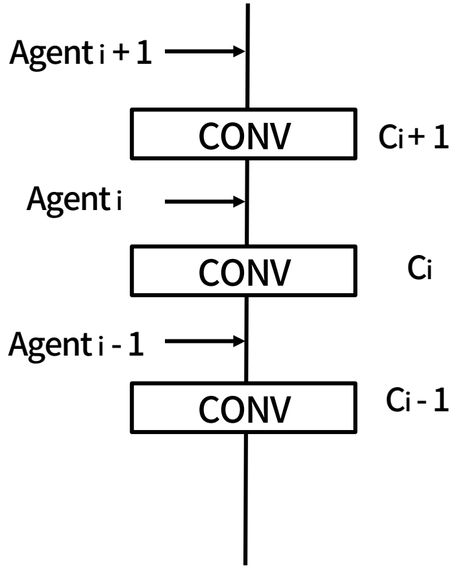
Bernoulli Sampling을 이용해 학습 중 낮은 중요도를
갖더라도 Agent를 통해 랜덤하게 사용 채널로 선택 가능

학습을 통해 중요한 채널을 선택하여 기존 모델과 같은
Optimizer, Loss Function 사용으로 상대적으로 cost가 낮음

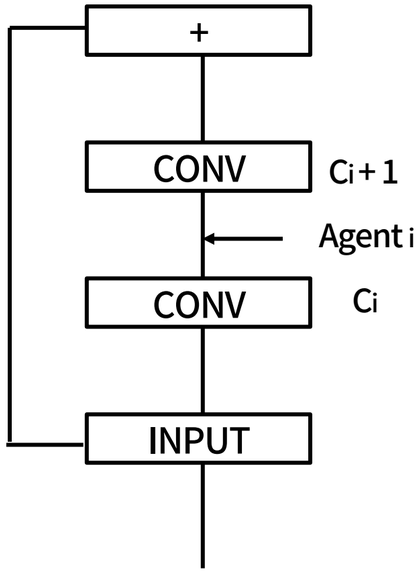
각 Agent에서 하나의 parameter를 학습해 빠르고 효율적

DECORE(Deep Compression with RL)

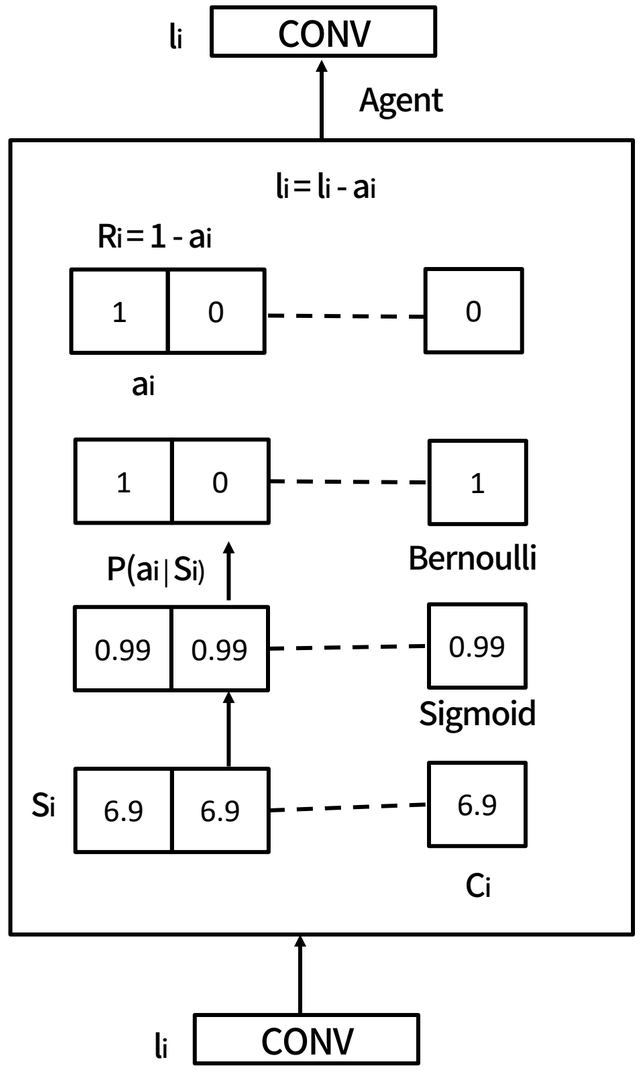
VGG



Resnet



Agent 정책

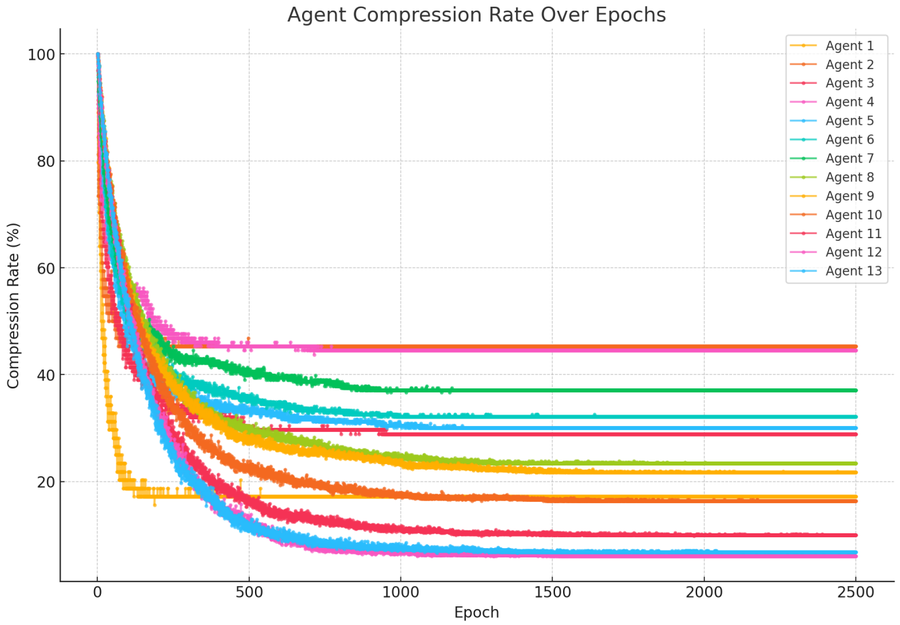


DECORE 구조

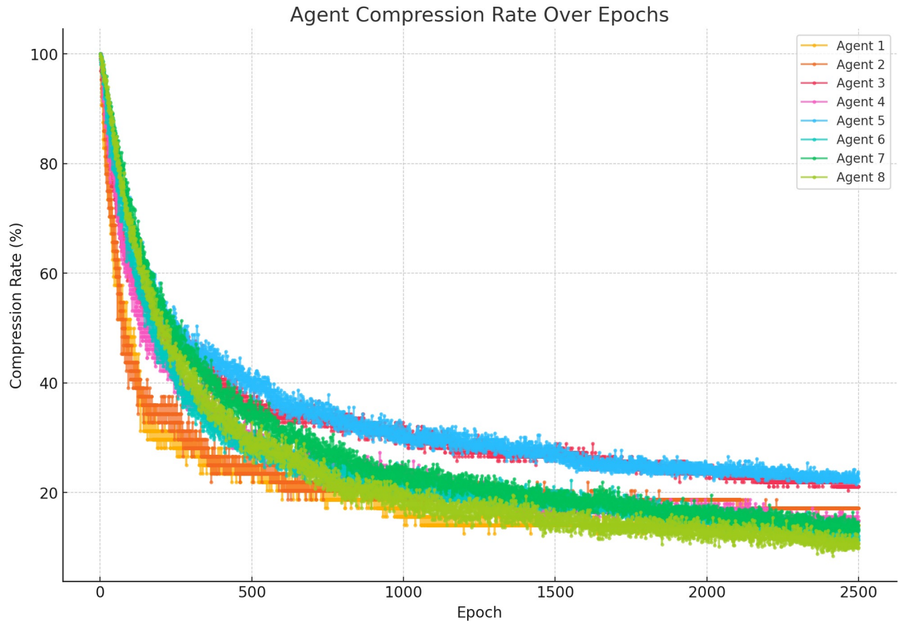
1. 각 Conv Layer마다 Agent 배치하여 Channel 별 유지/제거
   - VGG: 연속된 Conv Layer 사이에 Agent 배치
   - Resnet: Residual Block 내 Conv Layer 사이에 Agent 배치
2. MDP 구조에 따라 State(ex. 6.9) 정보를 수집하여 Sigmoid로 확률값 변환
3. 확률값 기반으로 Bernoulli Sampling 통해 0(제거) 또는 1(유지) 결정
4. 정확도 하락 시 Lambda값에 따라 음의 Reward를 주어 성능과 압축률 균형 유지
5. Reward 체계를 Loss Function 활용해 Policy Gradient 방식으로 Agent 정책 업데이트

DECORE(Deep Compression with RL)

VGG



Resnet



Unstructured Pruning 한계

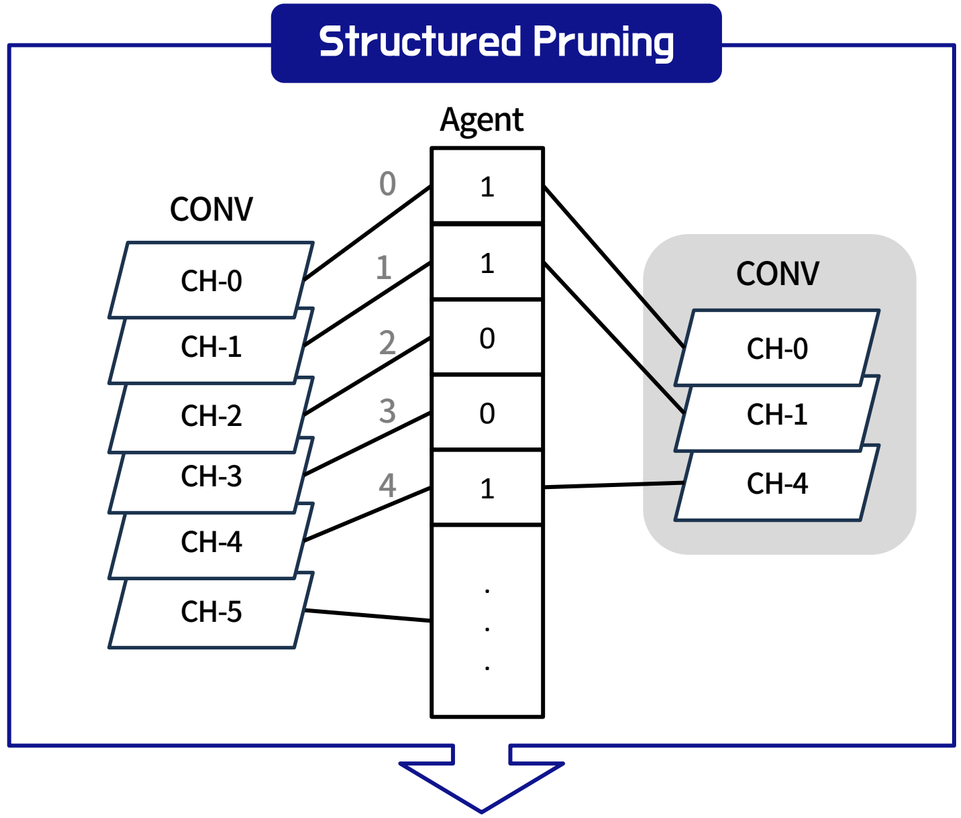
DECORE는 Unstructured Pruning을 하기 때문에 전체적인 네트워크 구조는 유지되고 모델 기본 용량이 크게 감소하지 않아

실제 하드웨어의 메모리와 연산량의 실질적 감소 제한적

GPU/TPU 같은 병령 처리 장치에 Unstructured Pruning으로 인해

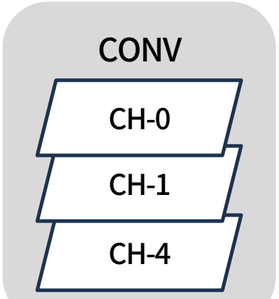
불규칙하게 분포된 가중치들이 연속적인 메모리 접근을 방해하고 캐시 효율성 저하시킴

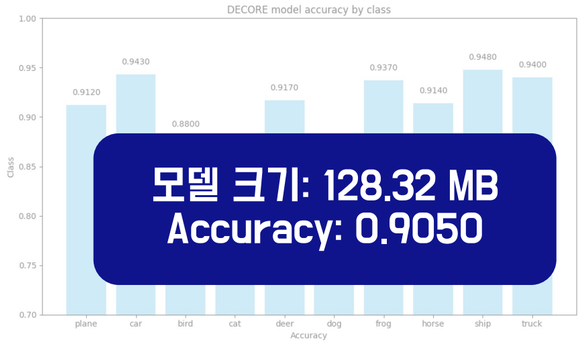
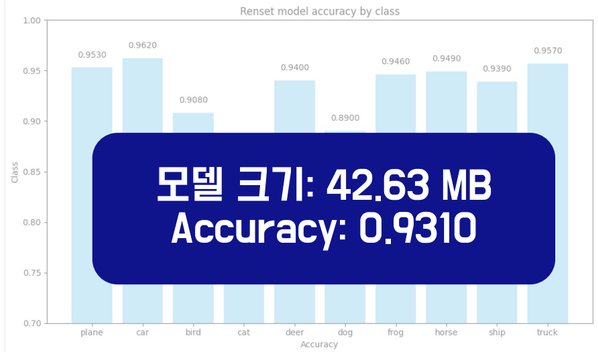
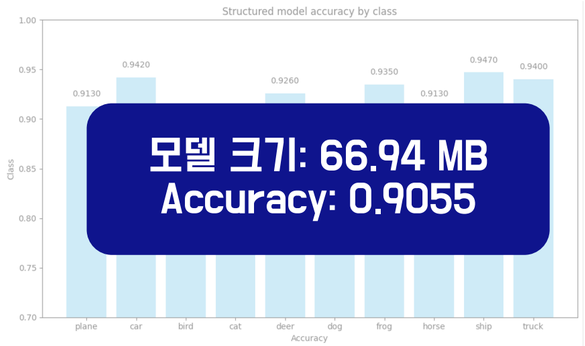
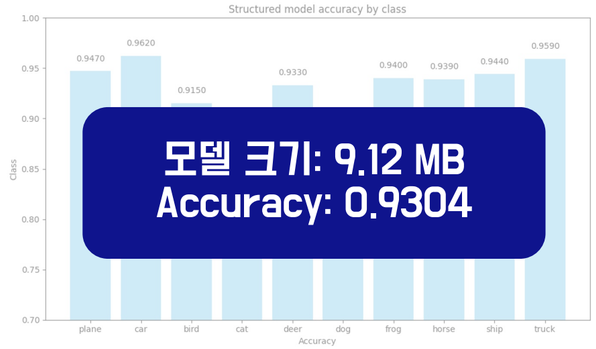
Structured Pruning



최종 Structured Pruning 된 Conv Layer

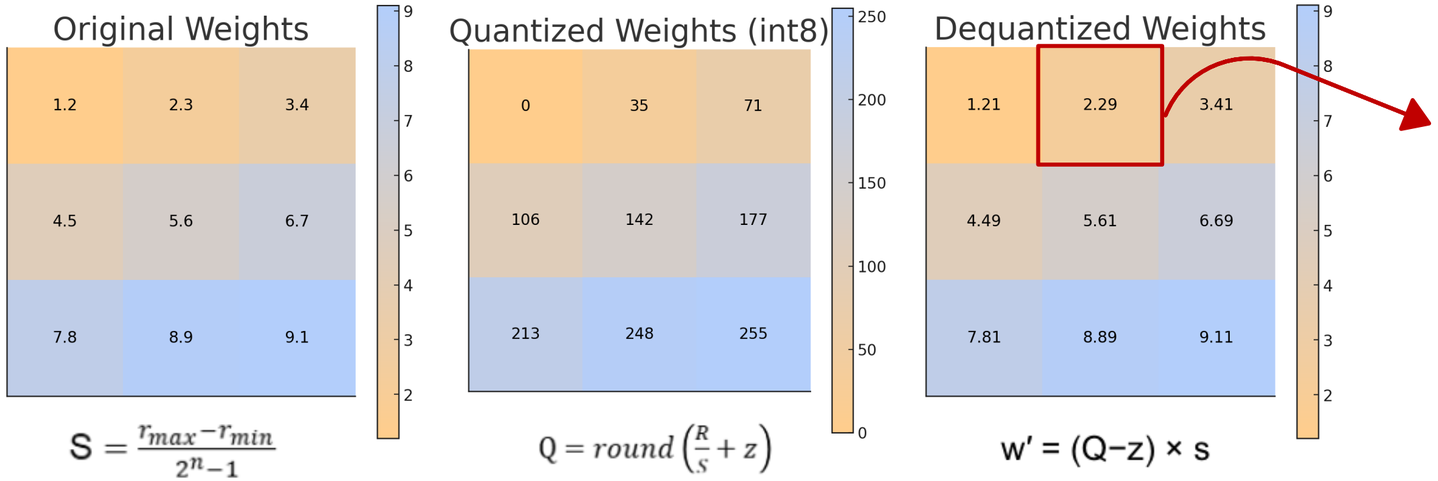
- Channel 수 ↓
- 용량 ↓



	VGG	Resnet
Unstructured	 모델 크기: 128.32 MB Accuracy: 0.9050	 모델 크기: 42.63 MB Accuracy: 0.9310
Structured	 모델 크기: 66.94 MB Accuracy: 0.9055	 모델 크기: 9.12 MB Accuracy: 0.9304

Quantizaion

1. Weight Quantization



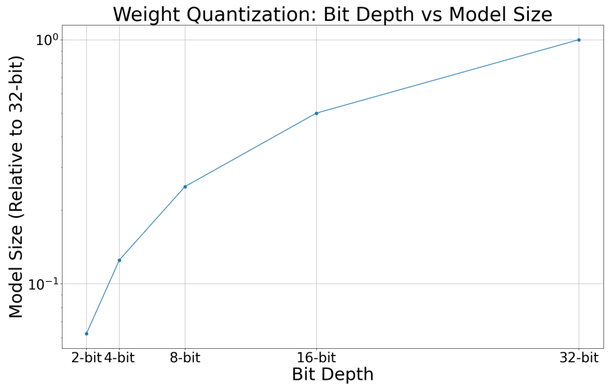
$$S = \frac{r_{max} - r_{min}}{2^n - 1}$$

$$Q = round\left(\frac{R}{S} + z\right)$$

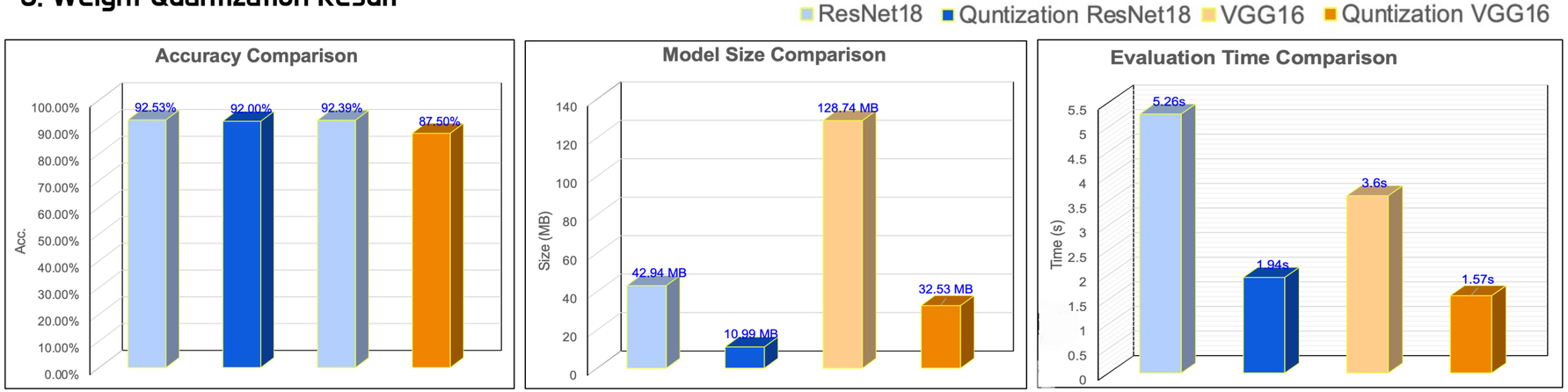
$$w' = (Q - z) \times s$$

- Scale = 0.03098 (값의 범위를 n-bit 정수로 정규화)
- Quantization: 35 = $round\left(\frac{2.3}{0.03098} - 39\right)$
- Zero Point(z): -39 = $round\left(\frac{\min(w)}{S}\right)$
- Dequantization: 2.2925 ≈ (35-(-39)) x 0.03098
- 기대효과: 데이터 압축률↑ 고속연산↑ 데이터 손실률↓

2. The Impact of Weight Quantization on Model Size



3. Weight Quantization Result



➡ Resnet18이 Quantization 후 모델 크기, 정확도 측면에서 더 효율적

Test Result

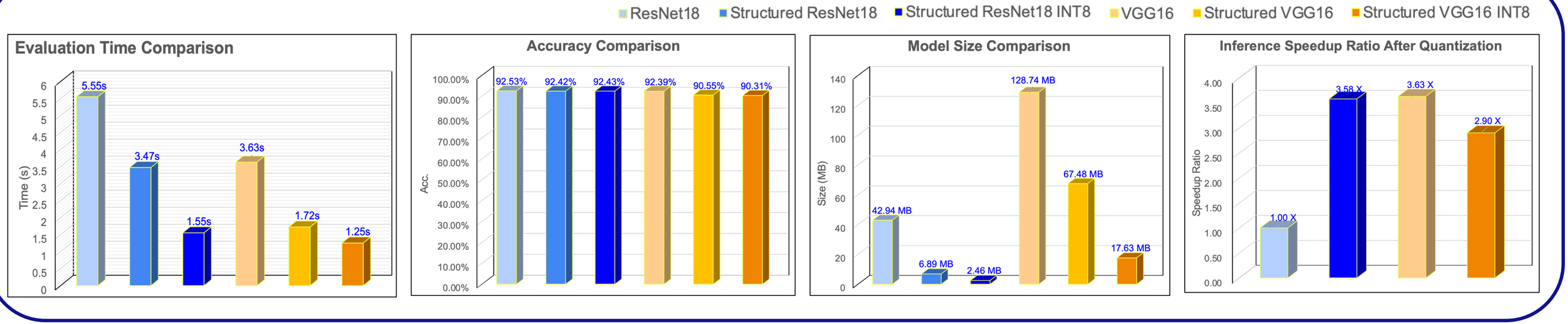
Model	Size (MB)	Top-1 Accuracy (%)	Precision (%)	Recall (%)	Evaluation Time (sec)	Accuracy Difference (%)
ResNet18	42.94	92.53	92.65	92.53	5.55	0
Structured ResNet18	6.89	92.42	92.47	92.42	3.47	-0.11
Structured ResNet18 INT8	2.46	92.43	92.48	92.43	1.55	-0.1
VGG16	128.74	92.39	92.38	92.39	3.63	0
Structured VGG16	67.48	90.55	90.59	90.55	1.72	-1.84
Structured VGG16 INT8	17.63	90.31	90.33	90.31	1.29	-2.08

Resnet18 Accuracy 차이는 최대 -0.1%로 영향이 거의 없고, Quantization과 Pruning후에도 높은 Accuracy 유지

VGG16 Quantization 후 정확도 감소가 미미하고, Test 시간에 대한 높은 효율성을 나타냄

Quantization Compression Analysis :Model Size Reduction (Original → int8)

- Resnet18: 42.94 MB ➡ 2.46 MB (40.48 MB 감소, 약 94.27% 감소)
- VGG16: 128.74 MB ➡ 17.63 MB (111.11 MB 감소, 약 86.31% 감소)



Conclusion

효율성 극대화

- 강화학습을 통해 모델 구조를 자동으로 탐색 및 최적화 가능
- 기존 휴리스틱 기반 압축 방법 대비 더 정교하고 효율적인 Pruning 전략 설계

성능-압축 균형 최적화

- 성능 저하를 최소화 하면서 경량화 모델 제공

새로운 AI 설계 패러다임

- 강화학습 활용으로 모델 설계 자동화를 넘어, AI 설계 지능화를 촉진
- 시간, 인적자원 절약 가능