

동적 저궤도 위성 네트워크에서 효율적 라우팅 연산자원 활용을 위한 AMP 기반 심층강화학습

조보근¹, 이현철^{*1}, 한명훈², 노봉수²

¹ 국립금오공과대학교 IT 융복합공학과

² 국방과학연구소

qhrmsgo@kumoh.ac.kr, *hcllee@kumoh.ac.kr

AMP-Based Deep Reinforcement Learning for Efficient Utilization of Routing Computation Resources in Dynamic LEO Satellite Networks

Bogeun Jo¹, Heoncheol Lee^{*1}, Myong-hun Han², Bong-Soo Roh²

¹ Department of IT Convergence Engineering, Kumoh National Institute of Technology

² Agency for Defense Development

요약

저궤도 위성 네트워크는 지상 인프라가 열악한 지역에서도 광범위한 커버리지와 초저지연 통신을 제공하지만 빠르게 변화하는 위성 토플로지와 제한된 온보드 컴퓨팅 자원으로 실시간 라우팅 응용이 어렵다. 본 논문에서는 PyTorch Automatic Mixed Precision(AMP)을 Dueling DQN 기반 심층강화학습 라우팅 모델에 적용하여 연산의 자원이 많이 소비되는 Convolution 과 Linear layer 를 FP16 으로, 수치 안정성을 위해 Loss 계산은 FP32 로 수행하는 혼합 정밀도 학습 기법을 제안한다. Loss scaling 을 통해 FP16 Gradient 의 underflow 를 방지하고, unscaling 후 FP32 로 Weight 를 안정적으로 재생산함으로써 학습의 안정성을 확보하였다. 제안된 AMP 기법은 single-precision(FP32) 모델 대비 성능은 유지하면서도 연산 자원 소비를 획기적으로 줄여 제한된 저궤도 위성 온보드 환경에서 실시간 학습 및 추론 요구를 충족시킴을 보였다.

I. 서론

최근 저궤도 위성 네트워크는 지상 인프라가 부족한 지역에서도 전 세계적으로 광범위한 커버리지와 초저지연 통신 서비스를 제공할 수 있어 차세대 글로벌 인터넷 인프라로 각광받고 있다. 그러나 위성의 지속적인 궤도 이동으로 인해 네트워크 토플로지가 자주 변동되며 우주 환경의 변화에 따라 위성 간 링크가 빈번히 단절되어 네트워크의 안정성을 유지하는 것이 매우 어렵다. 이러한 문제를 해결하기 위한 효과적인 방법으로 심층 강화학습을 활용한 라우팅 알고리즘이 주목받고 있다.

저궤도 위성은 우주 방사선, 급격한 온도 변화, 진동 등으로 인해 컴퓨팅 성능이 지상보다 제한적이다. 심층 강화학습 기법은 높은 성능의 연산을 요구하기 때문에 컴퓨터 성능이 제한적인 환경에서는 실시간 응용에 어려움이 있다. 이를 해결하기 위해 온보드 심층 강화학습이 필요하며 이는 동적으로 변하는 네트워크 환경에 신속히 대응하고 자원 제약을 줄여 높은 신뢰도와 성능을 위한 핵심 요소로 떠오르고 있다.

일반적으로 심층 강화학습 모델은 IEEE-754 single-precision(FP32) 연산을 사용하지만 FP32 는 32 비트 레지스터를 사용하므로 연산마다 많은 메모리와 전력을 소모한다. 이로 인해 위성의 제한된 전력, 컴퓨팅 자원을 빠르게 소모한다. 실제로 저궤도 위성에 탑재되는 온보드는 라우팅 외에도 다양한 작업을 병행해야 하므로, 라우팅 연산자원을 효율적으로 활용할 필요가 있다.

본 논문에서는 PyTorch Automatic Mixed Precision(AMP)을 적용하여 FP16 과 FP32 연산을 혼합하여 사용함으로써 심층강화학습 기반 저궤도위성 라우팅 모델에서 연산 자원을 효율적으로 활용하고자 한다.

II. 본론

1. 문제정의

저궤도 위성 네트워크는 여러 개의 위성이 궤도를 따라 고속으로 이동하므로 링크의 구조가 시간에 따라 불연속적으로 변화한다. 따라서 라우팅 문제는 수학적으로는 그리드 마르코프 결정 과정(Grid MDP, Grid-Markov Decision Process)으로 표현 할 수 있고, 이는 “first-order Markov assumption”이라는 가정에 기반하여 시간 t 에서의 상태는 $t-1$ 에서의 상태에만 영향을 받는다고 가정한다.

Grid MDP 는 상태 (State), 행동 (Action), 보상 (Reward), 상태 전이 확률 (State Transition Probability)로 구성된다. 각 위성의 상태좌표를 15×15 그리드 평면에 투영했고 상태는 각 위성의 위치와 해당 위성에 저장된 패킷의 정보를 나타내며, 행동은 각 위성에서 패킷을 전송하는 방향을 선택하는 것이다. 보상은 특정 방향으로 패킷을 전송했을 때 얻는 점수이며 각 이동당 -1, 목적지 도착 시 1 을 부여받는다. 상태 전이 확률은 패킷 전송 위치에 따른 위치 변화를 표현한다.

2. AMP 기반 Mixed-Precision Training 기법

본 연구는 PyTorch에서 제공하는 Mixed Precision Training[1] 기법을 구현한 Automatic Mixed Precision(AMP)를 활용하여 심층 강화학습의 연산을 동적으로 FP16 및 FP32로 전환했다. AMP는 Forward propagation에서 연산이 집중되는 Conv/Linear 연산을 FP16으로 실행하여 연산 속도를 높이고 메모리 사용량을 줄이며, 반대로 수치 안정성이 중요한 Loss 계산 및 최종 가중치(weight)저장과 같은 부분은 FP32 정밀도를 유지한다.

본 연구에서 사용된 심층 강화학습 모델인 Dueling DQN[2]은 상태 가치를 추정하는 스트림과 각 행동의 어드밴티지를 추정하는 스트림으로 구성되는데, 이 두 스트림 내부에 포함된 다수의 Conv2d, Linear, 그리고 활성화 함수 ReLU 등의 주요 연산들은 PyTorch AMP의 “with autocast():” 블록을 통해 자동으로 FP16으로 수행된다. 반면, Loss 계산은 모델 학습의 방향을 결정하는 중요한 단계이므로 수치적 안정성을 확보하기 위해 FP32 정밀도로 수행된다.

학습 과정에서 Gradient는 autocast 영역 내 연산의 영향으로 FP16으로 계산될 수 있는데, FP16은 표현 가능한 값의 범위가 좁아 매우 작은 Gradient 값이 0으로 소실되는 underflow 현상이 발생할 가능성이 있다. 이러한 문제를 해결하고 학습의 안정성을 확보하기 위해 “GradScaler”가 사용된다. “GradScaler”는 먼저 계산된 Loss 값에 동적으로 조절되는 scaling factor를 곱하여 그 크기를 인위적으로 확대한 후, 이를 기반으로 backpropagation를 진행하여 scale 된 FP16 Gradient를 얻는다. 이후, optimizer가 모델의 가중치를 갱신하기 직전에, 이 스케일된 Gradient를 다시 원래의 스케일로 unscaling(복원)하며, 이 과정에서 NaN이나 Inf와 같은 유효하지 않은 그래디언트가 발생하면 해당 배치의 가중치 업데이트를 건너뛰어 학습 안정성을 더욱 강화한다. 최종적으로 이렇게 복원되고 검증된 Gradient는 안정적인 FP32 형태로 변환되어 모델의 weight를 갱신하며, weight 자체도 FP32 형태로 저장되어 모델의 정밀도를 유지한다.

3. 학습 결과와 사용된 자원 비교 및 분석

본 연구는 i7-12700F, NVIDIA RTX 3070을 탑재한 PC에서 구현되었다. 표 1은 FP32 기반 Dueling DQN 모델과 AMP를 적용한 모델의 주요 학습 성능 지표를 정리한 것이며 그림 1은 학습 후 에피소드당 평균 score를 나타낸 그래프이다.

지표	FP32	AMP
평균 score	-36.9	-43.6
평균 훈련시간	57.45 ± 0.32 s	48.27 ± 1.17 s
평균 CPU 메모리	820 MB	928 MB
평균 GPU 메모리	34.7 MB	26.5 MB
평균 GPU 전력	131 W	70 W

표 1 학습 성능 지표 비교

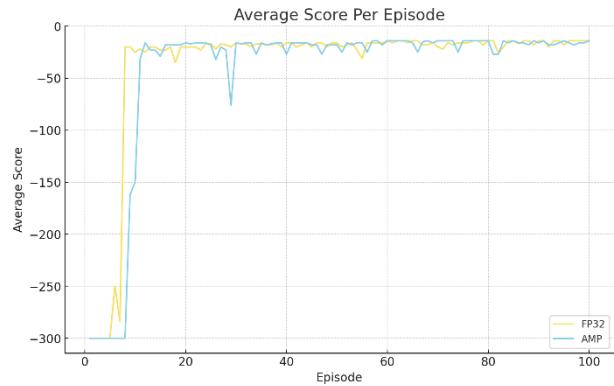


그림 1 Score 그래프 비교

두 모델 모두 100 에피소드 학습 후 누적 보상이 안정적으로 수렴하는 모습을 보였다. FP32 모델의 평균 score와 수렴 에피소드는 각각 약 -36.9, 8이며 AMP를 적용한 모델의 평균 score와 수렴 에피소드는 각각 약 -43.6, 12로 FP32 만을 사용한 모델이 학습 면에서는 더 안정적인 모습을 볼 수 있었다. 하지만 AMP를 적용한 모델 또한 후반부 에피소드에서 안정적으로 유지되어 AMP 적용에 따른 학습 손실은 경미함을 확인 할 수 있었다.

추가로, 학습 효율성과 연산 자원 사용량을 비교하였다. FP32 모델의 평균 훈련 시간은 57.45 ± 0.32 s, AMP 모델은 48.27 ± 1.17 s로 AMP를 적용한 모델이 약 16%빠른 것을 확인 할 수 있었다. FP32 모델의 평균 자원 사용량은 CPU, GPU 메모리 그리고 GPU 전력 각 약 820MB, 34.7MB, 131W였으며 AMP 모델은 각 928MB, 26.5MB, 70W로 CPU 메모리는 AMP를 적용한 모델이 오버헤드로 인해 약 13%정도 더 소비했지만 GPU 메모리와 전력은 약 24%, 47% 절감하며 AMP 기반 심층 강화학습 기법은 연산자원을 효과적으로 활용한 것을 알 수 있었다.

III. 결론

본 연구에서는 PyTorch AMP를 Dueling DQN 기반 저궤도 위성 라우팅에 도입해 FP16과 FP32 연산을 동적으로 혼합 처리함으로써 학습 시간을 16% 단축하고 GPU 메모리 24%, 전력 소모 47%를 절감하여 제한된 온보드 환경에서도 연산 효율을 크게 향상시킬 수 있음을 확인했다. 향후 연구에서는 INT16, INT8 저비트 양자화를 적용해 추가적인 연산 효율성을 달성하고자 한다.

ACKNOWLEDGMENT

이 논문은 2024년 정부(방위사업청)의 재원으로 국방과학연구소의 지원을 받아 수행된 연구임(UI247034TF).

참고 문헌

- [1] P. Micikevicius et al., "Mixed Precision Training," arXiv preprint arXiv:1710.03740v2, Oct. 2017.
- [2] 김도형, 이상현, 이현철, and 원동식, "동적 저궤도 위성 네트워크를 위한 Dueling DQN 기반 라우팅 기법," 대한임베디드공학호논문지, vol. 18, no. 4, pp. 173–183, 2023, doi: 10.14372/IEMEK.2023.18.4.173