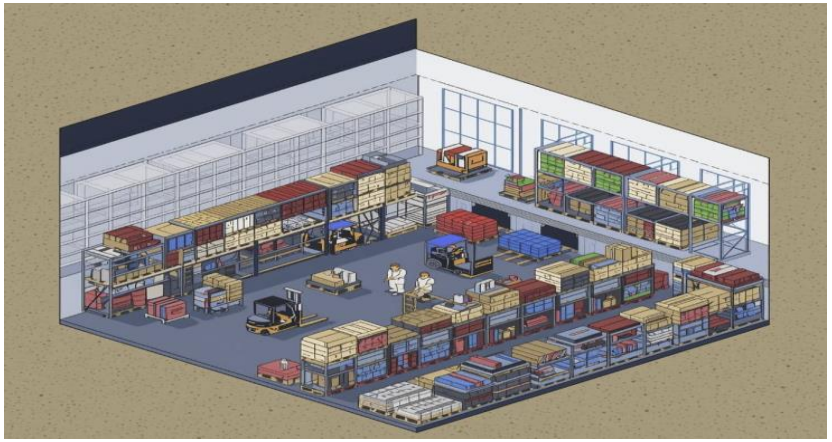
2025년도 2학기 서강대학교 AI·SW대학원

강화학습의 기초

『 불확실한 수요 환경에서의 적응형 재고 관리: Deep Q-Network 접근』



팀명: Q-Pang

A71035 어준성

A72063 유문식

https://github.com/Moong6/RL_Project

2025.12.7.

SOGANG UNIVERSITY

C O N T E N T S

목 차

SOGANG UNIVERSITY

Ⅰ. 프로젝트 주제 및 목표배경

연구 배경

홈쇼핑 현업 근무 경험을 통해 급격한 수요 변동에 따른 재고 관리의 구조적 한계를 인식함.
- 재고 과다(유지비 증가)와 재고 부족(매출 손실) 사이의 트레이드오프(Trade-off) 해결 필요성 증대.
- 기존 경험 의존적 방식에서 벗어나 데이터 기반의 자동화된 의사결정 시스템 필요

연구 목적

Double DQN을 활용하여 과대평가(Overestimation)를 방지하고 불확실한 수요 환경에서도
총 운영 비용(Total Cost)을 최소화하는 적응형 재고 관리 에이전트를 구현하고자 함.

연구 범위

Gymnasium 기반의 가상 물류 환경 구축 및 최적화 알고리즘 적용
- 데이터: 포아송 분포를 활용한 확률적 수요 및 주말 계절성 및 장기 추세(Trend) 반영
- 알고리즘: 학습 안정성을 높인 Double DQN 및 Experience Replay 적용
- 제약: 무작위 리드타임(입고 지연) 및 창고 용량 제약 반영

추진 방향

① MDP Modeling

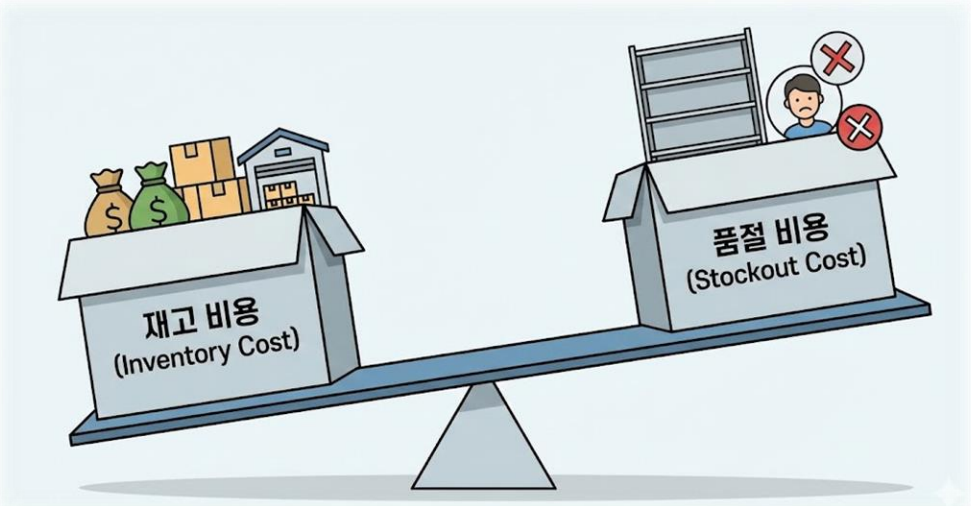
- 재고 및 파이프라인(입고 예정) 정보를 포함한 상태 정의
- 고정 주문비와 품절 패널티를 반영한 최적 보상 함수 설계

② 시뮬레이션 및 성능 평가

- 기존 재고 관리 기법 대비 비용 절감 효과 정량적 비교
- 패턴 시각화를 통해 선제적 대량 주문 등 학습된 전략의 유효성 검증

재고 관리의 이해

물류의 영원한 숙제는 '균형 ' 재고 과다(비용↑) vs 재고 부족(신뢰↓)



불확실한 수요, 왜 기존 방식으로는 안 되는가??

산업적 배경 (Motivation): 홈쇼핑/이커머스 특성상 트렌드뿐만 아니라 요일(주말), 이벤트에 따른 비선형적 수요 급증 패턴이 존재함

한계(Limitation): 기존의 고정 규칙((s, S) Policy)은 급변하는 수요와 계절성에 대응하기 어려움

목표(Goal): Double DQN을 활용하여 비용을 최소화하는 적응형 주문 정책(Adaptive Policy) 학습

Ⅰ. 팀 및 팀원 소개

팀명: Q-Pang (Q-learning + 쿠팡(이커머스/물류 대표 기업) 연상)

데이터사이언스·인공지능
어 준 성 (A71035)

프로젝트 기획 및 환경 모델링(MDP) 설계

데이터사이언스·인공지능
유 문 식 (A72063)

에이전트 구현 및 성능 최적화, 시뮬레이터 개발

"단순한 무작위(Random) 데이터로는 현실의 불확실성 을 학습할 수 없다."

데이터 생성 (Data Generation)
(실제 유통 데이터의 통계적 모사)

· 정적 데이터 대신 Poisson 분포를 활용하여 수요의 불확실성 모사
· 60일 주기의 계절성(Seasonality)과 장기적 수요 증가 추세(Trend)를 수식으로 구현
→ 복합적인 패턴이 섞인 고난이도 시뮬레이션 환경 구축

데이터 특징 (Data Features)
(계절성 및 주말 패턴 적용)

· 평일 vs 주말의 수요 격차를 인위적으로 조성
· 주말(토, 일) 수요가 평일 대비 1.3배(30%) 증가 하도록 설계
→ 에이전트가 요일별 변동성과 시계열 패턴을 스스로 인지하는지 검증

Ⅱ. MDP 정의(State, Action, Reward)

강화학습 에이전트가 환경과 상호작용하기 위한 MDP(Markov Decision Process)를 다음과 같이 정의함

State (St): 에이전트의 의사결정에 필요한 핵심 정보 4가지

- Inventory Level: 현재 창고 재고량
- Previous Demand: 전일 수요량 (추세 파악)
- Day of Week: 요일 정보 (주말 수요 급증 패턴 학습용)
- Pipeline Inventory: 재고뿐만 아니라 리드타임 동안 도착 예정인 물량을 상태에 포함하여 중복 주문 방지

Action (At): 주문 가능한 수량 (Discrete Space)

- A: {0, 10,000, 20,000, ..., 100,000} (대규모 창고 스케일 적용)

Reward (Rt): 비용 최소화를 위한 음의 보상 함수 설계

$$Rt = -(0.5 \times Inv + 50 \times Stockout + 1000 \times FixedCost + Purchase)$$

- 설계 의도

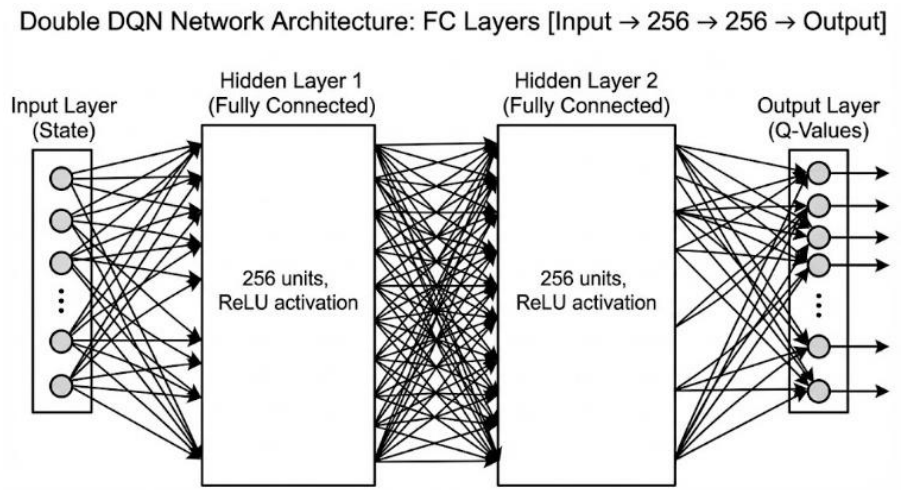
1. 품절 패널티를 유지비의 100배로 설정하여 품절 방지 강화 → 고객 신뢰 최우선
2. 고정 주문 비용을 반영하여 매일 조금씩 주문하는 것보다 최적 시점에 대량 주문(Batch Ordering)하도록 유도

알고리즘 및 Hyperparameter

Algorithm: Double DQN (DDQN)

· 일반 DQN의 Q-value 과대평가(Overestimation) 문제를 해결하여 학습 안정성 확보

Network: Input(State Dim) → FC(256) → ReLU → FC(256) → Output(Action Dim)



Hyperparameters

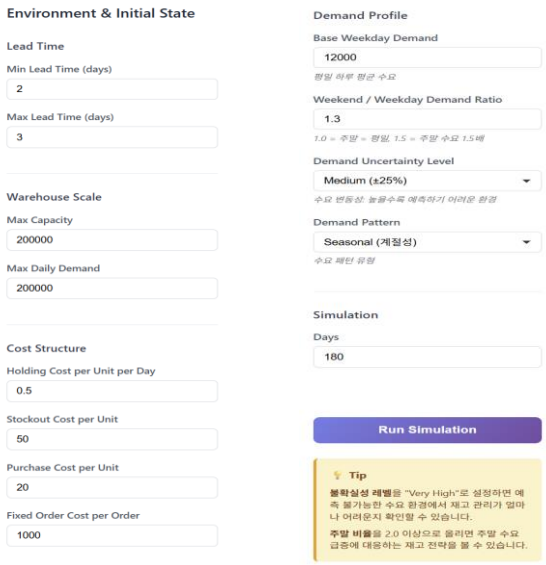
· γ = 0.99 (장기적 시야), lr = 1e-4 (안정적 학습)

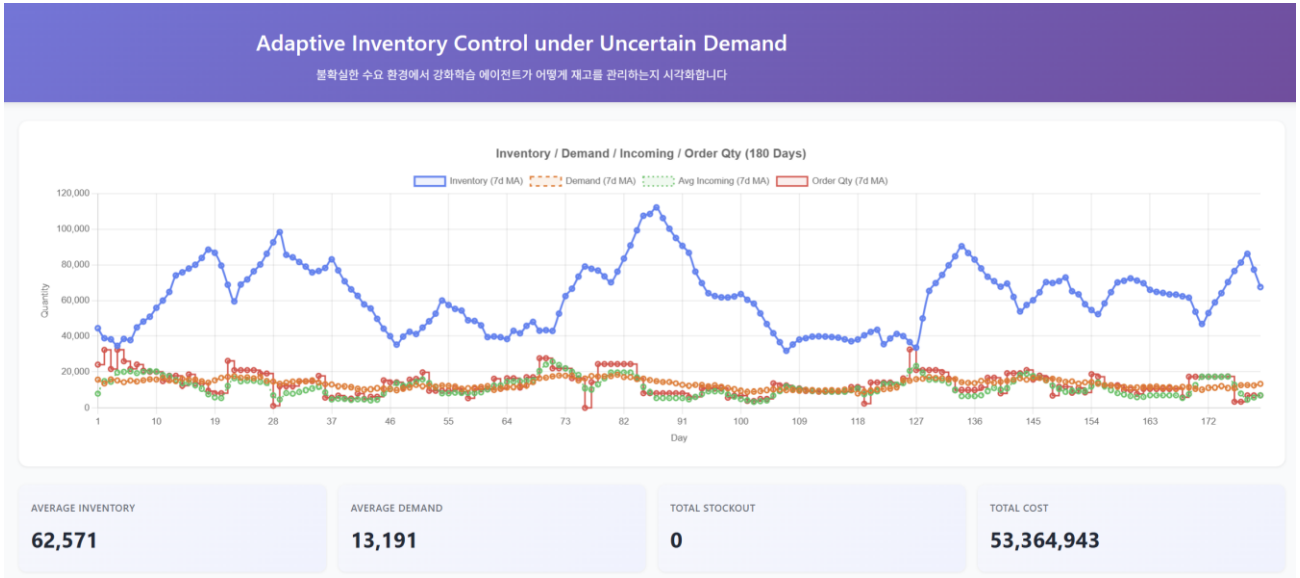
· Replay Buffer: 100,000, Batch Size: 256

· Epsilon Decay: 0.9998 (매우 천천히 탐험을 줄여 충분한 데이터 확보)

※ 핵심 파라미터(품절 비용 등) 조정 시 에이전트가 환경 변화를 인지하고 재고 운영 전략을 능동적으로 수정

사용자 정의 가능한 시뮬레이션 환경 (Customizable Simulation Environment)





1. 리드타임 & 스케일 (Logistics Constraints)

· Lead Time (Min/Max): 공급망 상황에 따른 입고 지연일 설정 가능

· Capacity: 창고 규모(최대 200,000개)에 따른 제약 조건 반영

2. 비용 구조 (Cost Structure)

· Holding vs Stockout Cost: 재고 유지비와 품절비 비율 조정을 통해 에이전트 성향 변화 실험 가능

· Fixed Order Cost: 고정 주문비 설정을 통해 배치 주문(Batch Ordering) 전략 유도

3. 수요 프로파일 (Demand Profile)

· Uncertainty Level: Low/Medium/High 등 변동성 강도 조절

· Demand Pattern: Seasonal(계절성), Trending(추세), Volatile(급변동) 등 다양한 수요 시나리오 테스트 지원

학습 성능 향상을 위한 하이퍼파라미터 최적화

Learning Rate (학습률) 조정

- **Before (1e-7)**: 학습 속도가 지나치게 느려 가중치 업데이트 미비 → **Underfitting 발생**
- **After (1e-4)**: 보폭을 키워 초기 수렴 속도를 확보하되 발산하지 않는 최적값 선정
- **Effect**: 학습 효율성 증대 및 정상적인 Loss 감소 유도

Epsilon Decay (탐험 감소율) 완화

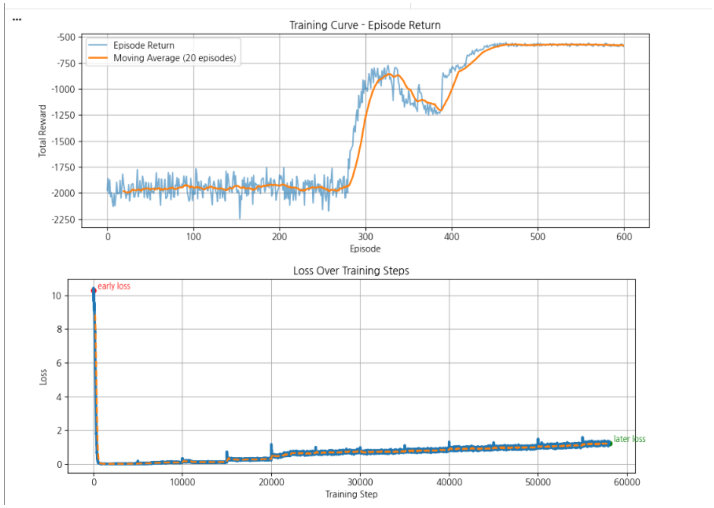
- **Before (0.999)**: 탐험(Exploration)이 너무 빠르게 종료됨 → **Local Optima(국소 최적해) 정체**
- **After (0.9998)**: 감쇠율을 완만하게 조정하여 충분한 탐험 기간(Exploration Phase) 확보
- **Effect**: State-Action Pair 경험을 통해 Global Optimum 수렴 유도

Replay Buffer Size 확대

- **Setting (100,000)**: 대용량 버퍼 메모리 확보
- **Purpose**: 데이터 간의 **시간적 상관관계(Temporal Correlation) 제거**
- **Effect**: 샘플의 **i.i.d(Independent and Identically Distributed) 조건**을 만족시켜 학습 안정성 강화

Ⅲ. 실험 결과 (1) - 학습 성능 분석 (Learning Curve)

코드 실행 결과 1: 학습 곡선(Reward) 및 Loss 그래프



[Phase 1: Exploration (0~280 ep)]: 높은 탐험률(ϵ)로 인해 무작위 주문이 발생하여 주문이 발생하며 −2,000 수준의 낮은 보상(높은 비용)에서 횡보

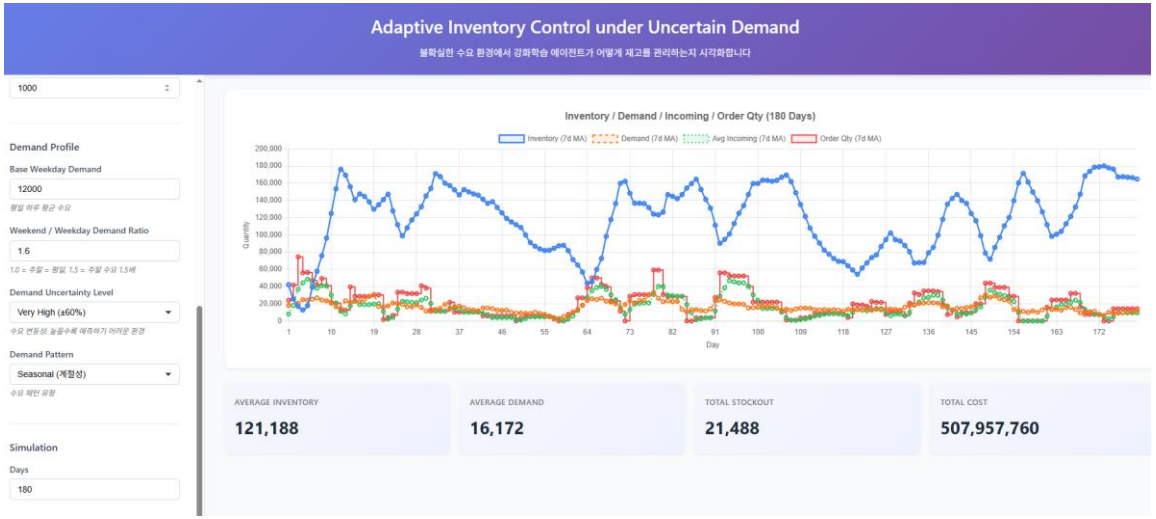
[Phase 2: Exploitation & Convergence (280~450 ep)]: 유의미한 패턴을 학습하며 보상이 급격히 상승. 약 300 에피소드 부근에서 성능의 퀀텀 점프(Quantum Jump) 발생

[Phase 3: Stability (450+ ep)]: 보상이 약 −600 수준에서 안정적으로 수렴. 초반 대비 약 70% 이상의 비용 절감 효과 달성

[Loss Analysis]: 초기 급락 후 완만하게 상승/유지되는 패턴은 Target Network가 지속적으로 업데이트되며 새로운 경험을 안정적으로 학습하고 있음을 시사함

Ⅲ. 실험 결과 (2) - 에이전트 행동 분석 (Behavior Analysis)

코드 실행 결과 2: 180일간의 Inventory/Demand/Order 그래프



[Strategy 1: High Safety Stock (안전 재고 확보)]
재고를 0에 맞추기보다 평균 수요 대비 넉넉한 재고를 상시 유지함
→ 품절 패널티(50)가 유지비(0.5)의 100배인 환경에서 품절 리스크를 원천 차단하려는 전략

[Strategy 2: Batch Ordering (대량 주문)]: 빨간색 그래프(주문)가 간헐적인 스파이크 형태로 나타남
→ 고정 주문비(1,000)를 절감하기 위해, 매일 주문하지 않고 필요한 시점에 몰아서 주문하는 경제적 판단

[Performance]
높은 수요 변동성(Variability)에도 불구하고, 재고 그래프(파란선)가 수요(주황선)의 추세를 안정적으로 방어함

Ⅳ. 결론 – 프로젝트 요약 및 시사점

Research Summary

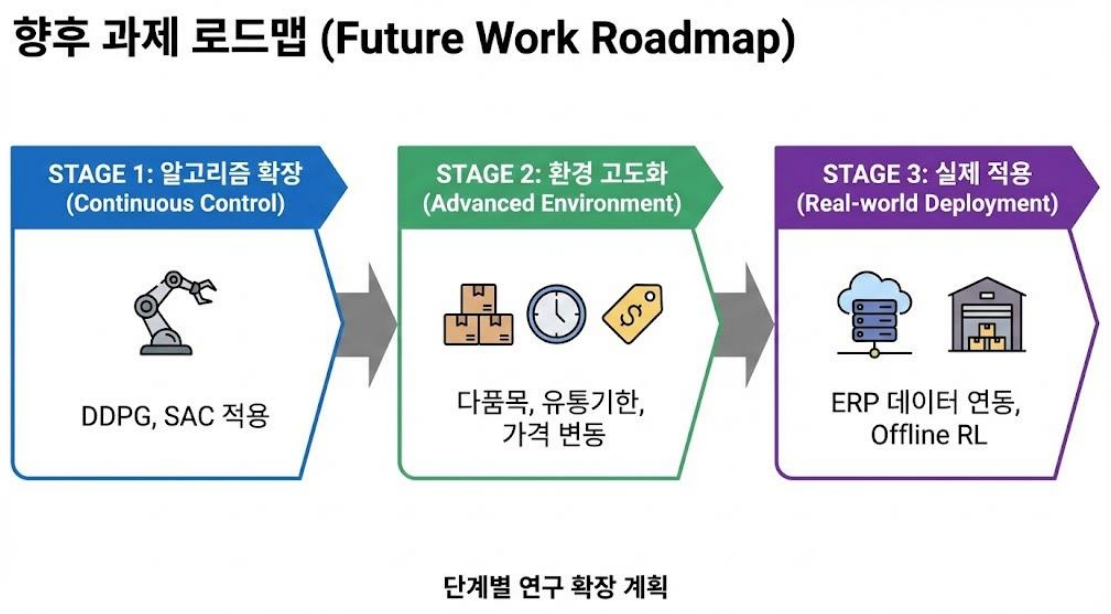
- 재고-품절-주문 비용 간의 **트레이드오프(Trade-off)**를 최적화하는 강화학습 환경 구축
- **Double DQN**을 통해 고정 주문 비용과 리드타임을 고려한 적응형 정책 학습 성공
- 학습 곡선 및 행동 패턴 분석을 통해 에이전트의 전략(대량 주문, 선제적 대응) 해석

Key Implications (시사점)

- **MDP 설계의 중요성**: 상태(State), 행동(Action), 보상(Reward) 설계가 알고리즘 자체보다 성능에 더 직접적인 영향을 미침
- **학습 안정성**: 보상 스케일링(Scaling)과 정규화가 학습 수렴의 핵심 요인임을 확인
- **가능성 확인**: 단순화된 환경이지만, RL이 복잡한 재고 관리 문제의 자동화된 정책 수립 도구로 활용될 가능성 입증

Limitatons (현실과의 괴리)

- **단일 품목(Single Item)** 및 단순화된 주문 단가 모델의 한계
- 이산화된 행동 공간(Discrete Action)으로 인한 정밀한 제어 부족



알고리즘 확장 (Continuous Control)

· DDPG, SAC 등 연속 제어 알고리즘을 도입하여 정밀한 주문량 최적화 시도

환경 고도화 (Multi-item & Pricing)

· 단일 품목을 넘어 다품목(Multi-item) 혼적 상황에서의 창고 용량 최적화

· 유통기한이 있는 신선식품(Perishable Goods) 시나리오 적용

Real-world Deployment

· 시뮬레이션 데이터가 아닌 실제 ERP 데이터와 연동하여 Offline RL 기법 적용 검토

감사합니다☺