

2024-08-04 Q-Table 설계 보고서

요약

1. 프로젝트 개요

목표: 구매자 역할을 수행하는 AI 에이전트가 판매자와의 협상에서 최대한 유리한 조건(낮은 가격)으로 거래를 성사시키도록 하는 강화학습 Q-Table 시스템 구축

핵심 제약:

- Q-Table 크기: 1,500개 셀 이하
- 학습 기간: 2~3개월 내 안정화
- 데이터 수집량: 월 5,000건 수준

2. State 변수 설계

2.1 설계 철학

- 표현력과 효율성 간의 균형 추구
- 협상 히스토리 대신 현재 상황에 집중하여 상태 공간 최적화

2.2 최종 State 변수 (총 36개 상태)

4분면 (시나리오 변수, S):

- A: 가격 최우선 → 공격적 가격 협상
- B: 중간 우세 → 가격 중시, 효율성 고려
- C: 균형 → 가격과 종료의 완전 균형
- D: 종료 중시 → 신속한 타결 우선

가격 구간 (P):

- PZ1: $P \leq A$ (목표가 이하) → 협상 종료 고려
- PZ2: $A < P \leq T$ (협상 가능 구간) → 적절한 압박
- PZ3: $P > T$ (임계값 초과) → 강력한 가격 압박

가격 수용률 (상대방 양보율):

- 낮음 (0-10%): 추가 압박 필요
- 중간 (10-25%): 적절한 양보 상태
- 높음 (25%+): 종료 시점 고려

2.3 Q-Table 크기 최적화

- 총 상태 수: $4 \times 3 \times 3 = 36$ 개
- Q-Table 크기: 36×9 (행동 수) = 324개 셀 (목표 대비 78% 절약)

3. 보상함수 설계

3.1 기본 구조

$$1 \quad R(s,a) = W \times (A/P) + (1-W) \times \text{End}$$

구성 요소:

- W: 동적 가중치 (시나리오 + 가격구간 반영)
- A/P: 가격 비율 보상 (목표가 대비 제안가)
- End: 협상 종료 보상

3.2 동적 가중치 시스템

$$1 \quad W = (S_n + PZ_n) / 2$$

시나리오 변수 (S_n):

- A: 1.0 (가격 절대 우선)
- B: 0.75 (가격 중시)
- C: 0.5 (균형)
- D: 0.25 (종료 우선)

가격 구간 변수 (PZ_n):

- PZ1: 0.1 (종료 고려)
- PZ2: 0.5 (균형 접근)
- PZ3: 1.0 (가격 압박)

4. 학습 방법론

4.1 CQL(Conservative Q-Learning) 기반 학습

손실함수:

$$1 \quad \text{Total Loss} = \text{Bellman Error} + \text{Conservative Penalty} + \text{Regularization}$$

핵심 구성:

- Bellman Error: 표준 Q-Learning 목적함수
- Conservative Penalty: 데이터에 없는 행동에 대한 과대평가 억제
- Regularization: 과적합 방지

4.2 하이퍼파라미터

- Conservative Factor (α): 0.5~2.0 (보수성 조절)

- Discount Factor (γ): 0.97
- Learning Rate: 0.0003~0.001 (적응적 조정)

5. 정책 및 탐험 전략

5.1 보수적 탐험 정책

- 안전 구간 내 제한적 탐험
- 시나리오별 차등 탐험 (A: 적극적, D: 최소한)

5.2 ϵ -greedy 변형

```
1  $\epsilon_{\text{adaptive}} = \epsilon_{\text{base}} \times \text{confidence\_factor} \times \text{scenario\_factor}$ 
```

6. 데이터 증강 - Dyna-Q Planning

6.1 작동 원리

1. **직접 학습**: 실제 데이터 1건 → Q-Table 1회 업데이트
2. **환경 모델링**: 실제 경험으로 가상 환경 구축
3. **계획**: 가상 환경에서 n회 시뮬레이션 → 추가 학습

6.2 효과

- 제한된 실제 데이터로 학습 효율성 극대화
- 월 5,000건 데이터로도 충분한 학습 가능

상세 설명

1. 개요

본 문서는 기업 간 협상 시뮬레이션을 위한 강화학습 에이전트의 Q-Table 설계 방안을 종합적으로 제시합니다. 본 프로젝트의 핵심 목표는 구매자 역할을 수행하는 AI 에이전트가 판매자와의 협상에서 최대한 유리한 조건(낮은 가격)으로 거래를 성사시키도록 하는 것입니다.

1.1 프로젝트 배경

현대 비즈니스 환경에서 협상은 기업의 수익성과 경쟁력을 결정하는 핵심 요소입니다. 특히 B2B 거래에서는 협상 능력이 직접적으로 비용 절감과 이익 창출로 이어지기 때문에, 체계적이고 일관된 협상 전략의 필요성이 증대되고 있습니다. 그러나 인간 협상자는 감정적 판단, 피로도, 경험 부족 등의 한계로 인해 항상 최적의 결과를 도출하기 어려운 상황에 직면합니다.

이러한 문제를 해결하기 위해 강화학습 기반의 AI 협상 에이전트 개발이 주목받고 있습니다. 강화학습은 환경과의 상호작용을 통해 최적의 행동 정책을 학습하는 기계학습 방법론으로, 협상과 같은 순

차적 의사결정 문제에 매우 적합한 접근법입니다.

1.2 설계 목표

본 Q-Table 설계의 핵심 목표는 다음과 같습니다:

주요 목표:

- 협상의 전략적 맥락과 현재 상황을 충분히 반영하여 AI가 최적의 행동을 결정하게 한다
- 한정된 기간(월 5,000건, 2~3개월) 내에 수집 가능한 데이터로 학습이 가능한 Q-Table 크기를 유지한다
- 가격 인하와 협상 종료 간의 최적 균형점을 찾는 보상 체계를 구축한다
- 실제 비즈니스 환경에서 활용 가능한 실용적인 솔루션을 제공한다

제약 조건:

- Q-Table 크기: 목표 1,500 셀 이하 (데이터 수집 현실성 고려)
- 학습 기간: 서비스 출시 후 2~3개월 내 안정화
- 데이터 수집량: 월 5,000건 수준의 협상 데이터

1.3 문서 구성

본 문서는 Q-Table 설계의 핵심 구성 요소들을 체계적으로 다룹니다. State 변수 설계에서는 협상 상황을 효과적으로 표현하는 방법을, 보상함수 설계에서는 AI의 학습 목표를 수학적으로 정의하는 방법을 제시합니다. 또한 CQL(Conservative Q-Learning) 기반의 학습 방법론과 Dyna-Q를 활용한 데이터 증강 기법을 통해 실제 구현 가능한 솔루션을 제공합니다.

2. State 변수 설계

2.1 설계 철학 및 제약사항

State 변수 설계는 Q-Table 기반 강화학습 시스템의 핵심 요소입니다. AI 에이전트가 현재 처한 협상 상황을 정확하게 인식하고, 이를 바탕으로 최적의 행동을 선택할 수 있도록 하는 것이 State 변수의 근본적인 역할입니다. 그러나 동시에 Q-Table의 크기가 기하급수적으로 증가하여 학습이 불가능해지는 상황을 방지해야 하는 현실적인 제약이 존재합니다.

핵심 설계 원칙:

첫째, 표현력(Expressiveness)과 **효율성(Efficiency)** 간의 균형을 추구해야 합니다. 협상의 복잡한 맥락을 충분히 담아내면서도, 현실적으로 학습 가능한 수준의 상태 공간을 유지해야 합니다. 이는 협상 도메인의 핵심적인 특성들을 식별하고, 이를 최소한의 변수로 압축하는 과정을 요구합니다.

둘째, 일반화 가능성(Generalizability)을 고려해야 합니다. 특정 협상 상황에만 적용되는 과도하게 세분화된 상태 정의보다는, 다양한 협상 시나리오에서 공통적으로 적용될 수 있는 추상화된 상태 표현을 선택해야 합니다.

셋째, 데이터 수집 현실성(Data Collection Feasibility)을 반영해야 합니다. 월 5,000건의 협상 데이터로 2~3개월 내에 안정적인 학습이 가능한 수준의 상태 공간을 설계해야 합니다. 이는 Q-Table의 각 셀당 최소 10개의 학습 데이터가 필요하다는 일반적인 강화학습 요구사항을 고려한 것입니다.

2.2 State 변수 선정 과정

초기 설계 단계에서는 협상의 모든 맥락을 포착하기 위해 '사용한 협상 카드의 순서(히스토리)'를 포함하는 방안이 검토되었습니다. 이는 협상의 진행 과정과 전략적 흐름을 직접적으로 반영할 수 있는 장점이 있었지만, 경우의 수를 기하급수적으로 증가시키는 치명적인 문제가 발견되었습니다.

구체적으로, 4개 시나리오 × 3개 가격구간 × 9^2 개 카드 조합의 경우 총 972개의 상태가 생성되며, 이는 9개 행동과 결합하여 8,748개의 Q-Table 셀을 요구합니다. 각 셀당 20개의 학습 데이터가 필요하다고 가정하면, 총 174,960개의 데이터가 필요하게 되어 현실적으로 학습이 불가능한 수준에 도달합니다.

이러한 문제를 해결하기 위해 협상의 핵심적인 맥락은 유지하면서도 상태 공간의 크기를 현실적인 수준으로 줄일 수 있는 대안적 접근법이 모색되었습니다. 그 결과, 다음 세 가지 핵심 변수를 중심으로 한 State 설계가 최종적으로 채택되었습니다.

2.3 최종 State 변수 구성

2.3.1 4분면 (시나리오 변수, S)

정의 및 역할:

4분면 변수는 협상의 전략적 목표와 우선순위를 정의하는 최상위 맥락 변수입니다. 이는 단순히 현재 가격 수준만을 고려하는 것이 아니라, 해당 협상에서 추구해야 할 근본적인 목표가 무엇인지를 AI 에이전트에게 명확히 전달하는 역할을 수행합니다.

시나리오별 특성:

시나리오 A (가격 최우선): 이 시나리오에서는 가격 인하가 절대적인 우선순위를 갖습니다. 협상 기간이 다소 길어지더라도 목표 가격에 최대한 근접한 결과를 도출하는 것이 중요합니다. AI 에이전트는 공격적인 가격 제시와 강력한 협상 카드 사용을 통해 지속적인 가격 압박을 가해야 합니다.

시나리오 B (중간 우세): 가격 인하를 우선시하되, 협상의 효율성도 함께 고려하는 균형잡힌 접근법을 요구합니다. 과도한 시간 소모 없이 합리적인 가격 수준에서 협상을 마무리하는 것이 목표입니다.

시나리오 C (균형): 가격 인하와 협상 종료 간의 완전한 균형을 추구합니다. 상황에 따라 유연하게 전략을 조정하며, 전체적인 협상 효과성을 극대화하는 것이 중요합니다.

시나리오 D (종료 중시): 신속한 협상 타결이 최우선 목표입니다. 시장 상황의 급변, 긴급한 조달 필요성, 또는 기회비용 등의 이유로 빠른 의사결정이 요구되는 상황에서 적용됩니다.

전략적 의의:

이러한 시나리오 구분은 AI 에이전트가 상황에 맞는 유연한 행동 선택을 가능하게 합니다. 동일한

가격 제안이라도 시나리오에 따라 전혀 다른 평가를 받을 수 있으며, 이는 보상함수의 가중치 조절을 통해 구현됩니다.

2.3.2 가격 구간 (인디케이터 가격 구간, P)

정의 및 역할:

가격 구간 변수는 현재 협상에서 제시된 가격이 구매자의 목표 및 수용 가능 범위와 비교하여 어느 위치에 있는지를 나타내는 핵심 지표입니다. 이는 AI 에이전트가 현재 협상의 진행 상황을 객관적으로 평가하고, 다음 행동의 강도와 방향을 결정하는 데 필수적인 정보를 제공합니다.

구간별 정의:

PZ1 ($P \leq A$): 제안 가격이 목표 가격 이하인 매우 유리한 상황입니다. 이 구간에서는 추가적인 가격 인하보다는 협상의 안정적인 마무리에 집중해야 합니다. 과도한 욕심으로 인해 협상이 결렬되는 위험을 방지하는 것이 중요합니다.

PZ2 ($A < P \leq T$): 제안 가격이 목표 가격과 수용 가능한 최대 가격 사이에 위치한 협상 가능 구간입니다. 이 구간에서는 적절한 수준의 가격 압박을 통해 목표 가격에 더 근접한 결과를 도출하려는 노력이 필요합니다.

PZ3 ($P > T$): 제안 가격이 수용 가능한 최대 가격을 초과하는 불리한 상황입니다. 이 구간에서는 강력한 협상 카드와 공격적인 전략을 통해 상당한 수준의 가격 인하를 이끌어내야 합니다.

전략적 활용:

가격 구간 정보는 보상함수의 가중치 조절에 직접적으로 활용됩니다. PZ1에서는 종료 보상의 비중이 높아져 협상 마무리를 유도하고, PZ3에서는 가격 보상의 비중이 높아져 지속적인 가격 인하 노력을 장려합니다.

2.3.3 가격 수용률 (상대방 가격 양보율)

정의 및 역할:

가격 수용률은 상대방(판매자)이 최초 제안 가격에서 현재까지 얼마나 가격을 양보했는지를 나타내는 지표입니다. 이는 초기에 고려되었던 '협상 카드 히스토리'를 대체하는 변수로, 협상의 진행 과정과 상대방의 협상 태도를 효율적으로 압축하여 표현합니다.

측정 방식:

가격 수용률은 다음 공식으로 계산됩니다:

$$1 \text{ 가격 수용률} = (\text{최초 제안가} - \text{현재 제안가}) / \text{최초 제안가} \times 100$$

예를 들어, 판매자가 최초에 1,000원을 제안했다가 현재 900원을 제안했다면, 가격 수용률은 10%가 됩니다.

구간별 분류:

- 낮은 수용률 (0-10%): 상대방이 가격 양보에 소극적인 상태

- **중간 수용률 (10-25%):** 상대방이 적절한 수준의 양보를 보이는 상태
- **높은 수용률 (25% 이상):** 상대방이 상당한 수준의 양보를 한 상태

전략적 의의:

이 변수는 AI 에이전트가 상대방의 협상 패턴과 양보 여력을 파악하는 데 핵심적인 역할을 합니다. 높은 가격 수용률은 상대방이 추가 양보 가능성이 낮음을 시사하므로 협상 종료를 고려해야 하며, 낮은 가격 수용률은 더 강력한 압박이 효과적일 수 있음을 의미합니다.

2.4 State 공간 크기 최적화

최종 State 변수 구성에 따른 상태 공간의 크기는 다음과 같이 계산됩니다:

- 4분면 (시나리오): 4가지
- 가격 구간: 3가지
- 가격 수용률 구간: 3가지

총 상태 수: $4 \times 3 \times 3 = 36$ 개

Q-Table 크기: 36×9 (행동 수) = 324개 셀

이는 목표했던 1,500개 셀보다 훨씬 작은 규모로, 현실적인 데이터 수집량으로도 충분히 학습 가능한 수준입니다. 각 셀당 10개의 데이터가 필요하다고 가정하면 총 3,240개의 데이터가 필요하며, 이는 월 5,000건의 수집 속도로 약 1개월 내에 확보 가능합니다.

2.5 State 표현 방식

실제 구현에서 State는 다음과 같은 형태로 표현됩니다:

```
1 State = (Scenario, PriceZone, AcceptanceRate)
2 예시: S(A, PZ2, AR1) = 시나리오 A, 가격구간 2, 수용률 구간 1
```

이러한 표현 방식은 코드 구현 시 명확성을 제공하며, 디버깅과 성능 분석을 용이하게 합니다. 또한 각 변수의 독립성을 보장하여 향후 시스템 확장이나 수정 시에도 유연성을 제공합니다.

3. 보상함수 설계

3.1 보상함수의 설계 철학

보상함수는 강화학습 에이전트의 학습 목표를 수학적으로 정의하는 핵심 요소입니다. 협상 도메인에서 보상함수 설계의 가장 큰 도전은 상충하는 두 가지 목표, 즉 '가격 인하'와 '협상 종료' 간의 최적 균형점을 찾는 것입니다. 과도하게 가격 인하에만 집중하면 협상이 결렬될 위험이 있고, 반대로 성급한 협상 종료에만 치중하면 충분한 가격 혜택을 얻지 못할 수 있습니다.

본 설계에서는 이러한 딜레마를 해결하기 위해 **동적 가중치 시스템**을 도입했습니다. 이는 협상의 맥락(시나리오)과 현재 상황(가격 구간)에 따라 두 목표 간의 중요도를 실시간으로 조절하는 혁신적인 접근법입니다.

3.2 기본 보상합수 구조

3.2.1 수학적 정의

에이전트가 상태 s 에서 행동 a 를 수행했을 때 받는 보상 $R(s,a)$ 는 다음과 같이 정의됩니다:

$$1 \quad R(s, a) = W \times (A / P) + (1 - W) \times \text{End}$$

변수 설명:

- **W**: 동적 가중치 ($0 \leq W \leq 1$)
- **A**: 구매자의 목표 기준 가격 (앵커링 목표가)
- **P**: 판매자가 제시한 현재 가격 (상대방 제안가)
- **End**: 협상 종료 여부 (종료 시 1, 진행 중 0)

3.2.2 보상 구성 요소 분석

가격 비율 보상 ($W \times A/P$):

이 항은 현재 제안 가격이 목표 가격에 얼마나 근접했는지를 평가합니다. A/P 비율이 1보다 클수록 (즉, 제안가가 목표가보다 낮을수록) 더 높은 보상을 제공합니다. 예를 들어, 목표가가 100원이고 제안가가 80원이라면 $A/P = 1.25$ 가 되어 기본 보상보다 25% 높은 가격 보상을 받게 됩니다.

종료 보상 ($(1-W) \times \text{End}$):

이 항은 협상이 성공적으로 완료되었을 때만 활성화됩니다. $(1-W)$ 값이 클수록 협상 종료에 대한 인센티브가 커지며, 이는 특히 목표 가격에 근접했거나 종료 중시 시나리오에서 중요한 역할을 합니다.

3.3 동적 가중치 시스템

3.3.1 가중치 계산 공식

동적 가중치 W 는 협상 상황과 가격 구간을 종합적으로 고려하여 다음과 같이 계산됩니다:

$$1 \quad W = (S_n + PZ_n) / 2$$

여기서:

- **S_n**: 시나리오 변수 (협상 전략에 따른 가중치)
- **PZ_n**: 가격 구간 변수 (현재 가격 위치에 따른 가중치)

3.3.2 시나리오 변수 (S_n) 설정

시나리오 변수는 협상의 전략적 우선순위를 반영하며, 값이 클수록 가격 개선이 더 중요함을 의미합니다:

시나리오	S_n 값	전략적 특성	적용 상황
A (가격 최우선)	1.0	가격 인하 절대 우선	예산 제약이 엄격한 경우

B (중간 우세)	0.75	가격 중시, 효율성 고려	일반적인 구매 상황
C (균형)	0.5	가격과 종료의 완전 균형	표준 협상 프로세스
D (종료 중시)	0.25	신속한 타결 우선	긴급 조달, 시장 변동성 높은 경우

3.3.3 가격 구간 변수 (PZ_n) 설정

가격 구간 변수는 현재 제안 가격의 상대적 위치를 반영하며, 값이 작을수록 목표 가격에 가까움을 의미합니다:

가격 구간	PZ_n 값	조건	전략적 의미
PZ1	0.1	$P \leq A$	목표 달성, 종료 고려
PZ2	0.5	$A < P \leq T$	협상 지속, 균형 접근
PZ3	1.0	$P > T$	강력한 가격 압박 필요

여기서 T는 수용 가능한 최대 가격 임계값을 의미합니다.

3.4 보상합수 동작 메커니즘

3.4.1 시나리오별 보상 패턴

시나리오 A (가격 최우선) 상황:

- PZ3에서: $W = (1.0 + 1.0) / 2 = 1.0$ → 완전히 가격 보상에 집중
- PZ2에서: $W = (1.0 + 0.5) / 2 = 0.75$ → 가격 보상 우선, 종료 보상 25%
- PZ1에서: $W = (1.0 + 0.1) / 2 = 0.55$ → 여전히 가격 우선이지만 종료 고려

시나리오 D (종료 중시) 상황:

- PZ3에서: $W = (0.25 + 1.0) / 2 = 0.625$ → 가격 보상 62.5%, 종료 보상 37.5%
- PZ2에서: $W = (0.25 + 0.5) / 2 = 0.375$ → 가격 보상 37.5%, 종료 보상 62.5%
- PZ1에서: $W = (0.25 + 0.1) / 2 = 0.175$ → 종료 보상이 82.5%로 압도적

3.4.2 실제 보상 계산 사례

다음은 구체적인 협상 상황에서의 보상 계산 예시입니다:

사례 1: 시나리오 A, 목표가 100원, 제안가 115원 (PZ2), 협상 진행 중

- $W = (1.0 + 0.5) / 2 = 0.75$
- $R = 0.75 \times (100/115) + (1-0.75) \times 0 = 0.75 \times 0.87 = 0.65$

사례 2: 시나리오 C, 목표가 100원, 제안가 83원 (PZ1), 협상 종료

- $W = (0.5 + 0.1) / 2 = 0.3$
- $R = 0.3 \times (100/83) + (1-0.3) \times 1 = 0.3 \times 1.20 + 0.7 \times 1 = 1.06$

사례 3: 시나리오 D, 목표가 100원, 제안가 100원 (PZ1), 협상 진행 중

- $W = (0.25 + 0.1) / 2 = 0.175$
- $R = 0.175 \times (100/100) + (1-0.175) \times 0 = 0.175 \times 1 = 0.175$

3.5 보상 확장 방안(협의 필요)

3.5.1 확장 보상 구조

기본 보상함수의 효과성을 더욱 높이기 위해 다차원 보상 체계로의 확장을 고려할 수 있습니다:

```
1 R_total = w1 × R_economic + w2 × R_relationship + w3 × R_strategic
```

경제적 가치 (R_economic):

```
1 R_economic = W × (A/P) + efficiency_bonus
```

여기서 efficiency_bonus는 협상 라운드 수를 고려한 효율성 보너스입니다.

관계적 가치 (R_relationship):

```
1 R_relationship = cooperation_score × satisfaction_factor
```

상대방과의 협력도와 만족도를 반영하여 장기적 파트너십을 고려합니다.

전략적 가치 (R_strategic):

```
1 R_strategic = scenario_achievement × position_strength
```

시나리오별 목표 달성도와 협상 포지션 강화 정도를 평가합니다.

3.5.2 시나리오별 가중치 조정

다차원 보상에서 시나리오별로 각 차원의 중요도를 다르게 설정할 수 있습니다:

시나리오	w1 (경제)	w2 (관계)	w3 (전략)	특성
A	0.7	0.2	0.1	경제적 이익 최우선

B	0.5	0.3	0.2	균형잡힌 접근
C	0.3	0.5	0.2	관계 중시
D	0.2	0.3	0.5	전략적 목표 우선

3.6 보상함수 검증 및 조정

3.6.1 보상 범위 정규화

CQL 학습의 안정성을 위해 보상 값의 범위를 [0, 1] 구간으로 정규화합니다:

```
1 R_normalized = (R - R_min) / (R_max - R_min)
```

3.6.2 보상 분산 최소화

시나리오별 보상 분산을 최소화하여 학습의 일관성을 확보합니다. 이는 특정 시나리오에서 과도하게 높거나 낮은 보상이 발생하는 것을 방지하여 균형잡힌 학습을 가능하게 합니다.

3.6.3 실험적 검증

보상함수의 효과성은 다음과 같은 지표들을 통해 검증됩니다:

- **수렴성:** Q값이 안정적으로 수렴하는지 확인
- **일관성:** 유사한 상황에서 일관된 행동을 보이는지 평가
- **효과성:** 실제 협상 성과(가격 달성률, 성공률)가 개선되는지 측정

이러한 종합적인 보상함수 설계를 통해 AI 에이전트는 복잡한 협상 상황에서도 최적의 의사결정을 수행할 수 있게 됩니다.

4. 학습 방법론

4.1 CQL 기반 학습 데이터셋 구축

4.1.1 데이터 구조 설계

CQL 학습을 위한 경험 튜플은 다음과 같은 구조로 설계됩니다:

```
1 Experience Tuple = (s, a, r, s', done)
```

구체적으로:

```
1 (CardID, Scenario, PriceZone, AcceptanceRate, NextCard, Reward, NextState, Terminal)
```

상태 정보 (s):

- CardID: 현재 사용 중인 협상 카드 식별자
- Scenario: 협상 시나리오 (A, B, C, D)

- PriceZone: 현재 가격 구간 (PZ1, PZ2, PZ3)
- AcceptanceRate: 상대방 가격 수용률 구간

행동 정보 (a):

- NextCard: 에이전트가 선택한 다음 협상 카드 (C1~C9)

보상 및 전이 정보:

- Reward: 보상함수로 계산된 즉시 보상
- NextState: 행동 수행 후의 다음 상태
- Terminal: 협상 종료 여부 (0 또는 1)

4.2 FQI + CQL 구현 세부사항

4.2.1 손실함수 설계

CQL의 핵심인 보수적 손실함수는 다음과 같이 구성됩니다:

```
1 Total Loss = Bellman Error + Conservative Penalty + Regularization
```

Bellman Error:

```
1 L_bellman = E[(Q(s,a) - y)²]
```

여기서 y 는 Bellman target으로:

```
1 y = r + \gamma \times (1 - done) \times \max_{a'} Q_{target}(s', a')
```

Conservative Penalty:

```
1 L_conservative = \alpha \times (E[Q(s,a) from \pi_{max}] - E[Q(s,a) from D])
```

여기서:

- α : Conservative factor (보수성 조절 하이퍼파라미터)
- π_{max} : 현재 정책에서 Q값을 최대화하는 행동 분포
- D : 실제 데이터셋의 행동 분포

Regularization:

```
1 L_reg = \lambda \times ||\theta||²
```

모델 파라미터의 과적합을 방지하기 위한 L2 정규화 항입니다.

4.2.2 하이퍼파라미터 최적화

Conservative Factor (α):

- 범위: 0.5 ~ 2.0
- 역할: 값이 클수록 더 보수적인 정책 학습

- 최적화 방법: 그리드 서치를 통한 실험적 결정

Discount Factor (γ):

- 설정값: 0.97 (기존 유지)
- 역할: 미래 보상의 현재 가치 할인율
- 협상의 단기적 특성을 고려하여 상대적으로 높은 값 설정

Learning Rate:

- 초기값: 0.0003
- 적응적 조정: 0.001까지 점진적 증가
- 스케줄링: 학습 진행에 따른 동적 조정

4.2.1 데이터 전처리 단계

1단계: 데이터 정제

- 결측값 처리: 불완전한 협상 기록 제거 또는 보간
- 이상값 탐지: 비정상적인 가격 제안이나 행동 패턴 식별
- 중복 제거: 동일한 상황의 중복 기록 정리

2단계: 특성 변환

- 범주형 변수 인코딩: 카드, 시나리오, 가격구간의 원-핫 인코딩
- 연속형 변수 정규화: 가격, 수용률 등의 Min-Max 정규화
- 시퀀스 데이터 구조화: 협상 진행 과정의 시계열 구조 반영

3단계: 데이터 분할

- 훈련 세트: 70% (모델 학습용)
- 검증 세트: 15% (하이퍼파라미터 튜닝용)
- 테스트 세트: 15% (최종 성능 평가용)

4.2.2 학습 과정 모니터링

수렴성 지표:

- Q값 변화량: 연속된 에포크 간 Q값 변화의 표준편차
- 손실 함수 추이: Bellman error와 conservative penalty의 개별 추적
- 정책 안정성: 동일한 상태에서의 행동 선택 일관성

성능 지표:

- 협상 성공률: 목표 가격 달성 비율
- 평균 가격 달성률: (목표가 / 최종가) × 100
- 협상 효율성: 평균 협상 라운드 수

조기 종료 조건:

- 검증 손실 증가: 연속 10 에포크 동안 검증 손실이 증가하는 경우
- Q값 발산: Q값이 비정상적으로 큰 값으로 발산하는 경우
- 정책 불안정: 동일 상태에서 행동 선택이 무작위적으로 변하는 경우

5. 정책 및 탐험 전략

5.1 정책 설계 원칙

협상 도메인에서의 정책 설계는 일반적인 강화학습 환경과는 다른 특수한 고려사항들을 요구합니다. 무엇보다도 실제 비즈니스 환경에서의 안전성과 예측 가능성이 핵심적인 요구사항이 됩니다.

5.1.1 보수적 탐험 정책

CQL 기반 학습에서는 기본적으로 오프라인 데이터에 기반한 보수적 정책을 채택합니다. 그러나 실제 배포 환경에서는 제한적인 수준의 탐험이 여전히 필요할 수 있습니다. 이를 위해 다음과 같은 계층적 탐험 전략을 제안합니다:

안전 구간 내 탐험: 데이터셋에서 충분히 검증된 상태-행동 조합 내에서만 제한적인 탐험을 허용합니다. 이는 Q값의 신뢰도 구간을 계산하여, 높은 신뢰도를 가진 행동들 중에서만 선택하는 방식으로 구현됩니다.

시나리오별 차등 탐험: 시나리오의 중요도에 따라 탐험 정도를 차등 적용합니다. 예를 들어, 시나리오 D(종료 중시)에서는 탐험을 최소화하고, 시나리오 A(가격 최우선)에서는 상대적으로 더 많은 탐험을 허용할 수 있습니다.

5.1.2 ϵ -greedy 변형 전략

전통적인 ϵ -greedy 전략을 협상 도메인에 맞게 변형하여 적용합니다:

```
1  $\epsilon_{\text{adaptive}} = \epsilon_{\text{base}} \times \text{confidence\_factor} \times \text{scenario\_factor}$ 
```

여기서:

- **ϵ_{base} :** 기본 탐험 비율 (0.05 ~ 0.1)
- **confidence_factor :** Q값 추정의 신뢰도에 따른 조정 인자
- **scenario_factor :** 시나리오별 위험 허용도 조정 인자

시나리오별 탐험 비율:

- 시나리오 A: $\epsilon \times 1.2$ (가격 최적화를 위한 적극적 탐험)

- 시나리오 B: $\varepsilon \times 1.0$ (표준 탐험)
- 시나리오 C: $\varepsilon \times 0.8$ (균형잡힌 보수적 접근)
- 시나리오 D: $\varepsilon \times 0.5$ (최소한의 탐험으로 안정성 우선)

5.2 실무진과의 협의 필요 사항

5.2.1 위험 허용도 설정

실제 비즈니스 환경에서의 탐험 정책은 다음과 같은 실무적 고려사항들을 반영해야 합니다:

거래 규모별 차등 적용: 고액 거래에서는 탐험을 최소화하고, 소액 거래에서는 상대적으로 더 많은 탐험을 허용하는 방식을 고려할 수 있습니다.

파트너 관계 고려: 신규 파트너와의 협상에서는 보수적 접근을, 기존 신뢰 관계가 구축된 파트너와는 상대적으로 유연한 접근을 취할 수 있습니다.

시장 상황 반영: 시장 변동성이 높은 시기에는 탐험을 줄이고, 안정적인 시기에는 새로운 전략을 시도해볼 수 있습니다.

6. Q값 초기화 전략

6.1 휴리스틱 기반 초기화

Q-Table의 초기값 설정은 학습 효율성과 수렴 속도에 큰 영향을 미칩니다. 무작위 초기화보다는 도메인 지식을 활용한 휴리스틱 초기화가 더 효과적입니다.

6.1.1 전문가 지식 기반 초기화

협상 전문가의 경험과 직관을 바탕으로 각 상태-행동 조합에 대한 초기 Q값을 설정합니다:

시나리오별 카드 선호도 반영:

- 시나리오 A에서는 공격적인 가격 협상 카드(예: 경쟁사 견적 제시, 대량 구매 조건)에 높은 초기값 부여
- 시나리오 D에서는 관계 유지 및 신속 타결 카드(예: 장기 계약 제안, 즉시 결정)에 높은 초기값 부여

가격 구간별 전략 반영:

- PZ1(목표가 이하)에서는 협상 종료 관련 행동에 높은 초기값
- PZ3(임계값 초과)에서는 강력한 가격 인하 요구 행동에 높은 초기값

6.1.2 LLM 기반 초기 데이터 생성

Large Language Model을 활용하여 다양한 협상 시나리오에 대한 초기 Q값을 생성합니다:

프롬프트 엔지니어링:

- 1 "다음 협상 상황에서 각 행동의 예상 효과를 1-10 점수로 평가해주세요:
- 2 - 현재 상황: 시나리오 A, 가격구간 PZ2, 수용률 중간

- 3 - 가능한 행동: [C1: 경쟁사 견적 제시, C2: 대량구매 조건, ...]
- 4 - 평가 기준: 가격 인하 가능성, 협상 성공 확률, 관계 유지"

7. 데이터 증강 기법 (Dyna-Q Planning)

7.1 Dyna-Q 기법 개요

Dyna-Q는 실제 경험(Real Experience)과 가상 경험(Simulated Experience)을 결합하여 학습 효율을 극대화하는 혁신적인 기법입니다. 협상 도메인에서 데이터 부족 문제를 해결하는 핵심 솔루션으로 활용됩니다.

7.1.1 기본 작동 원리

직접 학습 (Direct RL): 실제 협상 데이터 1건으로 Q-Table을 1번 업데이트합니다. 이는 기존의 전통적인 강화학습 방식과 동일합니다.

환경 모델링 (Model Learning): 실제 경험을 바탕으로 환경 모델을 구축합니다. 이 모델은 "특정 상태에서 특정 행동을 취했을 때 어떤 보상과 다음 상태가 나올 것인가"를 예측하는 가상 세계를 구현합니다.

계획 (Planning): 구축된 환경 모델을 활용하여 과거에 경험했던 상황들을 n번 반복 시뮬레이션하며 Q-Table을 추가로 업데이트합니다.

7.1.2 수학적 표현

Dyna-Q의 전체 학습 과정은 다음과 같이 표현됩니다:

```
1 For each real experience (s, a, r, s'):  
2     1. Direct Learning:  $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$   
3     2. Model Update:  $\text{Model}(s,a) \leftarrow (r, s')$   
4     3. Planning:  
5         For i = 1 to n:  
6              $s_{\text{sim}}, a_{\text{sim}} = \text{random\_previously\_experienced\_state\_action}()$   
7              $r_{\text{sim}}, s'_{\text{sim}} = \text{Model}(s_{\text{sim}}, a_{\text{sim}})$   
8              $Q(s_{\text{sim}},a_{\text{sim}}) \leftarrow Q(s_{\text{sim}},a_{\text{sim}}) + \alpha[r_{\text{sim}} + \gamma \max_{a'} Q(s'_{\text{sim}},a') - Q(s_{\text{sim}},a_{\text{sim}})]$ 
```

7.2 협상 도메인 특화 구현

7.2.1 환경 모델 설계

협상 환경의 특성을 반영한 환경 모델을 구축합니다:

상태 전이 모델: 현재 상태와 행동이 주어졌을 때 다음 상태의 확률 분포를 모델링합니다.

```
1  $P(s'|s,a) = \text{Multinomial}(\theta_{s,a})$ 
```

보상 모델: 상태-행동 조합에 대한 보상의 기댓값과 분산을 모델링합니다.

```
1  $R(s,a) \sim \text{Normal}(\mu_{s,a}, \sigma^2_{s,a})$ 
```

종료 확률 모델: 각 상태에서 협상이 종료될 확률을 모델링합니다.

```
1  $P(\text{done}|s,a) = \text{Bernoulli}(p_{s,a})$ 
```

7.2.2 Planning 전략 최적화

우선순위 기반 Planning: 모든 상태-행동 조합을 동일하게 시뮬레이션하는 대신, 중요도에 따라 우선순위를 부여합니다:

- **높은 우선순위:** 자주 방문되는 상태, 높은 보상을 가진 상태-행동 조합
- **중간 우선순위:** 최근에 업데이트된 상태, 불확실성이 높은 상태
- **낮은 우선순위:** 드물게 방문되는 상태, 안정적인 Q값을 가진 조합

적응적 Planning 횟수: 상황에 따라 Planning 횟수(n)를 동적으로 조절합니다:

```
1 n_adaptive = n_base × uncertainty_factor × importance_factor
```