|  |
| --- |
| **야구 투수의 최적 투구 자세를 위한 강화학습 기반 인공지능 모델 개발** |
| 배정훈, 부석준  경상국립대학교 컴퓨터과학과  {bjh0115, sj.buu}@gnu.ac.kr |
| Development of an artificial intelligence model based on reinforcement learning for the optimal pitching posture of baseball pitchers |
| Junghoon Bae, Seok-Jun Buu  Dept. of Computer Science, Gyeongsang National University |
| **요 약**  이 연구는 Soft Actor-Critic (SAC) 알고리즘과 Curriculum Learning을 활용하여 야구 투구 동작을 모방하는 강화학습 모델을 개발하고 최적화하는 데 중점을 두었다. SAC 알고리즘의 효율성과 견고함, 그리고 Curriculum Learning의 점진적 난이도 조절은 투구 동작의 모방에 있어 중요한 역할을 했다. 그러나 학습 과정에서는 Agent의 성능이 특정 지점에서 수렴하는 한계를 경험했다. 이는 Lesson 4에 해당하는 단계에서 임계 값에 도달하지 못하고 더 이상의 진전이 관찰되지 않은 것으로 나타났다.  이러한 한계는 향후 연구의 중요성을 강조한다. 더욱 다양한 학습 방식의 도입, hyper parameter의 재조정, 커리큘럼 구조의 개선 등이 필요할 것으로 보인다. 이를 통해 학습 모델의 효율성과 성능을 극대화하고, 더욱 정밀한 투구 모방과 최적화를 달성할 수 있을 것이다. 또한, 이 연구는 스포츠 과학 및 야구 분야뿐만 아니라, 실시간 의사결정이 중요한 다양한 응용 분야에서 강화학습의 적용 가능성을 탐구하는 기반이 될 수 있다. 따라서, 본 연구의 결과와 한계는 강화학습을 활용한 미래의 연구와 응용에 중요한 기초 자료를 제공한다. | | |

**1. 연구 배경 및 문제 정의**

강화학습과 인공지능 기술은 스포츠 과학 분야에서 중요한 역할을 하고 있으며, 이는 특히 야구와 같은 분야에서 선수들의 성능 향상에 기여하고 있다. Soft Actor-Critic (SAC) 알고리즘은 이러한 기술 중 하나로, 연속적인 행동 공간에서의 효율적인 학습을 가능하게 하며, 이를 통해 선수들의 의사결정과 기술을 최적화할 수 있다. 본 보고서는 KBO 리그 투수들의 국제 경쟁력을 강화하기 위해 SAC 알고리즘을 중심으로 한 강화학습 모델을 최적화하는 방법에 대해 탐구한다.

Unity 환경과 ML-Agent를 활용한 시뮬레이션을 통해, 투수들의 투구 기술을 실제와 유사한 조건에서 분석하고 개선하는 방법을 모색할 것이다. 이러한 접근은 투구 속도와 정확도의 향상을 목표로 하며, 국제 대회에서의 성공 가능성을 높이는 데 중요한 역할을 할 것으로 기대된다. 강화학습의 효과적인 적용은 스포츠 분야에서 실시간 의사결정의 중요성을 강조하며, 이는 특히 야구와 같은 실시간 반응이 중요한 스포츠에서 매우 중요하다.

본 논문은 SAC 알고리즘의 내부 메커니즘을 분석하고, 이를 바탕으로 강화학습 모델을 최적화하는 방안을 제시한다. 유니티 환경에서 구현된 ML-Agent를 사용한 투구 모델을 통해 이 알고리즘의 실험적 평가를 수행하며, 하이퍼 파라미터 및 환경 설정의 개선 방향을 탐색할 예정이다. 이 연구는 강화학습을 적용하는 다양한 분야에 중요한 시사점을 제공할 것이며, KBO 리그의 국제 경쟁력을 강화하는 데 중요한 기여를 할 것으로 기대된다.

**2. 관련 연구**

강화학습은 인공지능 분야에서 복잡한 환경 내에서의 최적의 의사결정 문제를 해결하는 주요 기법 중 하나이다. 이 분야는 다양한 알고리즘들의 발전을 통해 혁신을 거듭해왔으며, 이들 알고리즘은 각각의 특정 환경과 목표에 맞게 설계되었다.

대표적인 강화학습 알고리즘으로는 Q-learning, Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), 그리고 Soft Actor-Critic (SAC) 등이 있다. 이 중 Q-learning과 DQN은 행동 가치 함수를 학습하는 데 중점을 두는 반면, PPO와 같은 정책 최적화 방법은 직접적인 성능 최적화를 목표로 한다.

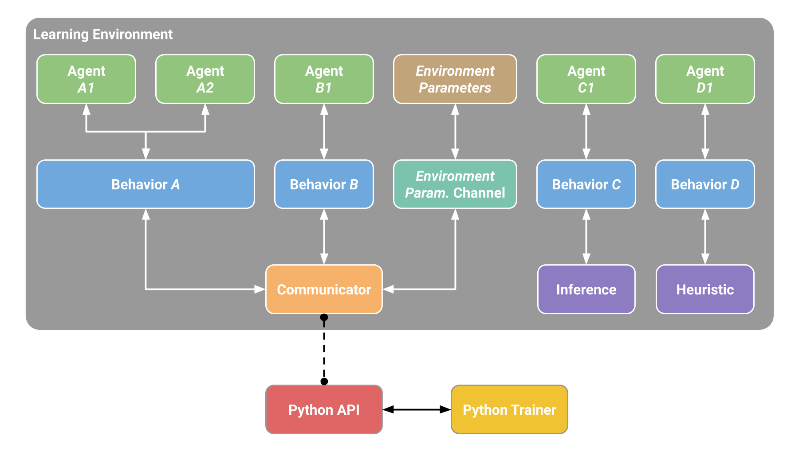
본 연구에서는 연속적인 행동 공간과 복잡한 환경에서 뛰어난 성능을 발휘하는 SAC 알고리즘을 선택했다. SAC는 최대 엔트로피 강화학습 원칙을 기반으로 하며, 이를 통해 탐색 능력과 견고한 학습을 동시에 달성한다. 이는 스포츠 과학과 야구 투구와 같은 분야에서 복잡한 의사결정과 기술적 전략을 최적화하는 데 특히 유리하다[3].

Unity는 고도의 시뮬레이션을 구축할 수 있는 강력한 플랫폼으로, 실제 환경을 모사하는 데 필수적인 요소를 제공한다. ML-Agent는 이러한 Unity 환경에서 강화학습 알고리즘을 구현하고 실험하는 강력한 도구이다. ML-Agent의 구조는 에이전트가 환경에서 관찰한 데이터를 바탕으로 행동을 결정하고, 수행한 행동에 따른 결과(보상)를 학습하는 프로세스를 지원한다. 이러한 구조는 복잡한 스포츠 과학 응용에서 실시간 데이터 처리와 지속적인 학습이 필요한 경우에 특히 중요하다[4].

이러한 접근법은 스포츠 과학에 AI를 접목하는 새로운 방향을 제시하며, 본 연구는 야구 투수의 투구 기술과 전략 개선을 위한 구체적인 사례로서 중요한 기여를 할 것으로 예상된다.

**3. 방 법**

**3.1. ML-Agent**

ML-Agents는 Unity Technologies에서 개발한 기계 학습 프레임워크로, 주로 비디오 게임 및 시뮬레이션 환경에서 인공 지능(AI) 모델을 개발하고 훈련시키는 데 사용된다. 이 프레임워크는 복잡한 환경에서의 에이전트 행동 학습, 신경망 기반의 결정 프로세스, 그리고 실시간 상호 작용 시뮬레이션에 초점을 맞추고 있다. ML-Agents는 다양한 학습 알고리즘을 지원하며, 본 고에서는 SAC 학습 알고리즘을 통한 학습을 진행한다. 그림 1은 ML-Agent의 원리를 간단하게 표현한 예시 상황 에서의 내용이다.

[그림 1] block diagram of ML-Agent

ML-Agent의 구성 요소는 [그림1]과 같다.

첫째, '환경(Environment)'은 에이전트가 상호 작용하는 가상의 공간으로, 다양한 장애물과 목표를 포함한다. 둘째, '에이전트(Agent)'는 이 환경 내에서 독립적으로 행동하며, 센서를 통해 정보를 수집하고 이를 바탕으로 행동을 결정한다. 셋째, '옵저버(Observer)'는 에이전트가 수집한 정보를 처리하여 인공 신경망에 입력한다. 이 정보는 시각적 또는 수치적 데이터일 수 있다. 넷째, '신경망(Neural Network)'은 입력 데이터를 바탕으로 최적의 행동을 예측한다. 이 신경망은 경험을 통해 지속적으로 최적화된다. 다섯째, '보상 시스템(Reward System)'은 에이전트가 긍정적인 결과에 대해 보상을 받고 부정적인 결과에 대해서는 패널티를 받는 방식으로 작동한다. 여섯째, '학습 알고리즘(Learning Algorithm)'은 에이전트의 경험을 바탕으로 신경망을 조정하고 개선한다. 일곱째, '트레이너(Trainer)'는 전체 학습 과정을 관리하며, 에이전트의 성능을 모니터링하고 학습 과정을 조정한다. 이러한 구성 요소들의 상호 작용을 통해 ML-Agents는 에이전트가 환경과의 지속적인 상호 작용을 통해 시간이 지나면서 더 나은 결정을 내리는 방법을 습득하도록 지원한다.

**3.2. SAC 알고리즘**

SAC의 핵심은 Actor-Critic 구조를 기반으로 하며, 여기서 ‘Actor’는 에이전트의 행동을 결정하고, ‘Critic’은 이러한 행동의 가치를 평가한다. 이 구조는 행동 선택과 가치 평가가 병행되어 효율적인 학습이 가능하도록 한다.

이 알고리즘의 주요 특징 중 하나는 Soft Policy Update이다. 이는 에이전트가 보상을 최대화하는 것뿐만 아니라, 탐험에 대해서도 보상을 받으며, 따라서 더욱 다양한 행동을 시도하고 최적의 전략을 찾아갈 수 있도록 한다.

SAC는 최대 엔트로피 강화학습(Maximum Entropy Reinforcement Learning) 원칙을 따른다. 이 원칙은 단순히 보상을 최대화하는 것을 넘어서, 정책의 엔트로피를 높여 에이전트가 더욱 탐험적이고 견고한 학습 행동을 보이게 하는 것을 목표로 한다.

샘플 효율성 측면에서도 SAC는 주목할 만한 알고리즘이다. 복잡한 환경에서 학습을 진행할 때 필요한 데이터 양이 상대적으로 적다는 특성을 가지고 있어, 다양한 응용 분야에서 유용하게 활용될 수 있다.

SAC 알고리즘은 그 특성상 자율 주행 차량의 제어, 로봇 팔의 정밀한 움직임 제어, 복잡한 시뮬레이션 환경에서의 전략 개발 등과 같은 다양한 분야에서 응용될 수 있다. 이러한 분야에서는 연속적인 행동 공간과 고도의 정밀성이 요구되며, SAC는 이러한 요구사항을 효과적으로 충족시키는 강력한 도구이다.

**3.3 Curriculum Learning**  
 Curriculum Learning은 강화학습의 효율성을 증진시키는 핵심 전략 중 하나로, 인간이 새로운 기술을 학습할 때 단계별로 점진적으로 난이도를 증가시키는 방식을 모방한다. 이 방법은 특히 복잡하고 어려운 작업을 다루는 경우에 유용하며, 훈련 과정을 여러 단계로 분할하여 각 단계마다 특정 학습 목표를 설정한다.

본 프로젝트에서는 Curriculum Learning을 활용해 Agent의 투구를 유도한다. 투수로부터 18.44m 거리에 있는 스트라이크 존으로 투구하는 것을 유도하기 위해, 18가지의 Lesson으로 세분화함으로써 요구되는 시간을 효과적으로 단축할 수 있을 것으로 예상한다.

**4. 실험 및 결론**

**4.1 Simulation 환경**

본 보고서에서 제안하는 강화학습 시뮬레이터는 실제 야구 투구 메커니즘을 상세하게 모사하기 위해 Unity3D 환경에서 설계되었다. 시뮬레이션의 설계는 투수 에이전트의 신체 모델링부터 환경 요소에 이르기까지 여러 단계에 걸쳐 체계적으로 이루어졌다.

텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명

[그림 2]

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

[그림 3]

Simulation 환경에서는 인간의 동작을 재현하는 Humanoid 3D 모델을 사용하여 가상의 야구 투수를 구현한다. [그림 2]에서 Simulation 환경을 설명한다.

Agent는 인간의 생체역학적 특성을 반영하여 설계된 3D Humanoid 모델로 구성된다. 인간의 신체를 총 30가지의 BodyPart로 재현하였다. 각 BodyPart는 [그림 3]과 같이 ConfigurableJoint 컴포넌트를 통해 Agent가 실제 인간의 동작 범위와 유사하게 움직일 수 있도록 한다[5]. Rigidbody 컴포넌트를 통해 강체를 재현하여 중력과 같은 다양한 물리적 현상을 재현할 수 있도록 한다. 뿐만 아니라, 각 BadyPart에 적합한 Collider 컴포넌트를 통해 충돌체를 구현하고 충돌과 관련된 현상을 재현한다. 해당 BodyPart가 지면과 맞닿아 있는지 여부를 실시간으로 검사하는 GroundContact 컴포넌트는 Agent가 두 발로 지면에 서있도록 유도하기 위해 추가되었다.

Brain의 Observer는 30개 관절의 지면 접촉 여부를 확인하고, Hip part를 제외한 29개 관절의 회전값과 현재 가해지는 힘을 관찰한다. 또한 야구공과 스트라이크 존의 3차원 위치 정보를 관찰한다. Agent의 Action은 29개의 관절의 회전방향 및 가해지는 힘을 결정한다. 총 181차원 벡터에 대한 관측을 매 Step마다 입력받고, 116개의 ContinuousAction을 출력한다. 이를 수신한 Agent의 JointDriveController 컴포넌트가 action들을 각 관절에 실제 물리적인 동작으로 구현한다.

Reward/Penalty 규칙은 다음과 같이 구성되었다.

1. 야구공이 Agent의 손과 맞닿아 있는 매 단위시간마다, 단위시간에 비례한 Penalty를 부과한다. 이 때, Penalty는   
   과 동일하다.
2. 야구공이 지면과 닿을 때, 스트라이크 존과의 수평거리에 따라 0~5의 Reward를 지급한다. 5에 가까울수록 스트라이크 존과 가까움을 의미한다.
3. 야구공이 스트라이크 존과 닿을 때, Reward를 10 지급하고, 추가로 야구공의 스트라이크 존 방향으로 갖는 속도에 비례한 보상을 지급한다.
4. GroundContact 컴포넌트를 포함한 BodyPart 중 Foot을 제외한 BodyPart가 지면에 닿는 순간, 1의 Penalty를 지급한다.
5. Head의 높이가 Hip의 높이보다 작거나 같아지는 순간, 50의 Penalty를 지급한다.

규칙 1)은 Agent가 공을 손에서 놓도록 유도하기 위해 정했다.

규칙 2)와 3)은 Agent가 성공적인 투구를 할 수 있도록 유도하기 위해 정한 규칙이다. 규칙 4)와 5)는 두 발로 지면을 딛고 서 있을 수 있도록 하면서도, 투구 자세가 무너지지 않도록 하기 위해 정한 규칙이다.

**4.2. 학습**

본 프로젝트의 학습은 앞서 서술한 것처럼, SAC알고리즘과 Curriculum Learning을 활용하여 진행한다. 조절 가능한 hyper parameter 및 학습과 관련된 설정은 다음과 같이 진행하였다.

trainer\_type: sac

hyperparameters:

learning\_rate: 0.0003

   learning\_rate\_schedule: constant

   batch\_size: 1024

   buffer\_size: 20000

   buffer\_init\_steps: 0

   tau: 0.005

   steps\_per\_update: 30.0

   save\_replay\_buffer: true

   init\_entcoef: 1.0

   reward\_signal\_steps\_per\_update: 30.0

 network\_settings:

   normalize: true

   hidden\_units: 512

num\_layers: 3

vis\_encode\_type: simple

goal\_conditioning\_type: hyper

deterministic: false

reward\_signals:

extrinsic:

gamma: 0.995

strength: 1.0

keep\_checkpoints: 10

checkpoint\_interval: 1000000

max\_steps: 10000000

time\_horizon: 1000

Curriculum Learning을 활용하는 방안으로는, 19개의 Lesson을 정의하였다. Agent가 복잡한 관절들을 이용해 야구공을 18.44m까지 던지는 것은 상당히 어렵고 복잡한 동작이기 때문에, Agent로부터 1m, 2m, … 와 같이 초반의 Lesson에는 가까운 거리에 스트라이크 존을 배치하고, Lesson이 진행됨에 따라 스트라이크 존이 점점 멀어져 마지막 Lesson에는 실제 야구장의 조건과 같이 18.44m에 스트라이크 존이 배치되도록 했다. 다음은 이와 같은 방식으로 학습을 진행한 모델의 통계이다.

[그림 4] Cumulative Reward

[그림 5] Distance\_Between\_Ball\_and\_StrikeZone

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명[그림 4]은 학습 중인 모델의 누적보상을 나타내는 표이며, [그림 5]는 야구공이 지면에 닿는 순간 스트라이크 존과의 수평 거리를 나타낸다. 0에 가까울수록, 스트라이크 존과 가까우며, 1보다 큰 경우에는 스트라이크 존의 정 반대 방향의 지면에 낙하함을 의미한다. 스트라이크 존에 도달하지 않고 받을 수 있는 최대 Reward는 5이므로, Cumulative Reward 5를 초과하는 것은, 드물게 나마 스트라이크 존에 야구공이 도달함을 의미한다. 5를 넘어 상승할수록, 그 빈도가 잦아짐을 의미하여, 그 임계 값이 7.5를 초과하면 다음 Lesson으로 넘어가기에 Step 2.6M, Step 3.6M, Step 7M을 기점으로 다음 Lesson으로 넘어 감에 따라 전후로 잠시동안 급격한 변화를 보인다.다음 [그림 6]은 학습을 통해 구성된 신경망에 대한 구조를 나타낸 도식이다.

[그림 6] 신경망의 구조

**4.3. 결론 및 향후 연구**

본 프로젝트에서는 야구 투구 동작을 모방하기 위해 Curriculum Learning을 도입했습니다. 초기 단계에서 이 방법은 Agent가 의도된 행동을 어느 정도 따르게 하는 데 성공적이었다. 그러나 학습 과정의 중반인 8M Step에서부터 [그림 4]와 [그림 5]에 나타난 바와 같이 에이전트의 성능이 특정 지점에 수렴하며 더 이상의 진전이 관찰되지 않았다. 이는 Lesson 4에 해당하는 부분으로, Reward가 설정된 임계 값에 도달하지 못해 다음 단계로 진행하지 못한 것으로 보인다. 이는 현재의 학습 방법에 대한 한계를 나타내며, 예정된 19개의 Lesson을 모두 수행하지 못한 점은 보완해야 할 점으로 보인다.

향후 연구에서는 본 프로젝트에서 나타난 수렴 문제를 해결하기 위해 또 다른 학습 방식을 도입하거나, hyper parameter 설정 및 커리큘럼 구조를 재검토하고 조정할 필요가 있는 것으로 보인다.

**참고 문헌**

[1] "Reinforcement Learning: An Introduction", Richard S. Sutton and Andrew G. Barto, MIT Press, 2018.

[2] "Unity ML-Agents Toolkit", Unity Technologies, [Online available: <https://github.com/Unity-Technologies/ml-agents>].

[3] Haarnoja, T., et al. (2018). "Soft Actor-Critic Algorithms and Applications." arXiv preprint arXiv:1812.05905. [Online available: <https://arxiv.org/abs/1812.05905>].

[4] Juliani, A., et al. (2018). "Unity: A General Platform for Intelligent Agents." arXiv preprint arXiv:1809.02627. [Online available: <https://arxiv.org/abs/1809.02627>].

[5] Cocchiarella, Linda. AMA(미국의사협회) 장해평가법 가이드. 서울: 중앙문화, 2006.