[붙임 2]

경 북 대 학 교  인 공 지 능 혁 신 융 합 대 학 사 업 단

## AICOSS 자율주행로봇 경진대회

# 결과보고서

| | | | |
|---|---|---|---|
| **팀명** | 애드 아스트라 | **팀장** | 곽유열 |
| | | **팀원** | 최문성 |
| **소속대학** | 성균관대학교<br>서울과학기술대학교<br>경북대학교 | **팀원** | 최동혁 |
| | | **팀원** | 이원겸 |

2026.01.30.

# 1. 데이터 및 코드 제출 항목 체크

■ 산학플랫폼에 코드 및 데이터 업로드가 완료되었는지 확인했습니다.

# 2. 개발 과정

## 2.1 자율주행

### 2.1.1 문제 해결 전략

본 해커톤의 자율주행 과제는 카메라 영상 기반 주행 제어 문제로 정의할 수 있으며, 차량은 주행 중 획득한 전방 영상을 바탕으로 조향 및 속도에 해당하는 연속적인 제어 값을 실시간으로 예측해야 한다. 특히 8자 형태의 트랙은 직선 구간과 급격한 곡선 구간이 반복적으로 등장하며, 바닥의 녹색 카펫과 흰색 라인이라는 제한된 시각적 단서를 기반으로 안정적인 주행이 요구된다.

이에 본 팀은 해당 문제를 이미지 입력 → 연속 제어값 출력의 회귀(regression) 문제로 설정하고, End-to-End 방식의 딥러닝 모델을 활용한 해결 전략을 수립하였다. 즉, 별도의 규칙 기반 제어 또는 특징 추출 단계를 두기보다는, 카메라 이미지로부터 주행에 필요한 핵심 시각적 패턴을 직접 학습한다.

문제 해결의 핵심 전략은 다음과 같다. 첫째, 불필요한 시각 정보 제거를 통한 입력 데이터 정제이다. 실제 주행에 중요한 정보는 트랙 바닥과 라인에 집중되어 있으므로, 이미지의 상단부(천장, 관중, 배경 등)를 제거하고 하단부 중심의 ROI(Region of Interest)를 설정하였다. 둘째, 연속 제어에 적합한 신경망 구조 설계이다. 조향 및 속도 값은 이산적인 클래스가 아닌 연속 값이므로, 분류 모델이 아닌 회귀 모델을 채택하였다. 셋째, 실제 차량 제어 특성을 고려한 출력 스케일링 전략이다. 출력 값은 실제 서보모터 및 구동계에 적용되므로, 안정적인 범위를 유지하도록 정규화 및 활성화 함수를 설계하였다.

### 2.1.2 모델 개발

(1) 데이터 수집 및 전처리

학습 데이터는 Digital 8 트랙에서 수집된 주행 이미지와, 해당 시점의 주행 제어 값이 파일명에 포함된 형태로 제공되었다. 이미지 파일명에는 두 개의 실수 값이 포함되어 있으며, 이는 각각 조향 및 속도(또는 좌·우 주행 제어값)에 해당한다.

이미지는 OpenCV를 통해 불러온 후, 세로 방향으로 [120:270] 범위만을 사용하여 ROI를 설정하였다. 이를 통해 주행과 직접적인 관련이 없는 배경 요소를 제거하고, 모델이 트랙 라인과 바닥 질감에 집중할 수 있도록 유도하였다. 최종 입력 데이터의 형태는 (150, 400, 3)으로 정의되었다. 레이블 값은 실제 제어 범위를 고려하여 400으로 나누어 정규화하였으며, 출력층에서 sigmoid 활성화 함수를 사용하여 모델 출력이 0과 1 사이에 위치하도록 설계하였다.

(2) 모델 구조 및 학습 전략

본 모델은 Residual CNN 구조를 기반으로 설계되었다. 다수의 Conv2D 계층을 통해 저수준 시각적 특징(라인, 경계, 질감)부터 고수준 특징을 점진적으로 추출하며, 중간에 잔차 연결(Residual Connection)을 추가함으로써 깊은 네트워크에서 발생할 수 있는 정보 손실 및 학습 불안정을 완화하였다.

활성화 함수로는 ReLU 대신 Swish 함수를 사용하였다. Swish는 연속적인 출력이 요구되는 회귀 문제에서 보다 부드러운 함수 형태를 제공하여, 조향 각도와 같은 미세한 제어 값 예측에 유리하기 때문이다.

출력층은 Dense(2) 구조로 구성되었으며, 이는 두 개의 연속 제어 값을 동시에 예측하기 위함이다. 손실 함수로는 Mean Squared Error(MSE)를 사용해 예측 값과 실제 값 간의 오차를 직접적으로 최소화했다.

모델 학습에는 Adam Optimizer(learning rate = 1e-3)를 사용하였으며, 과적합 방지를 위해 Early Stopping 기법을 적용하였다. 검증 손실이 20 epoch 동안 개선되지 않을 경우 학습을 조기에 종료한다.

### 2.1.3 모델의 실제 적용과 목표 대비 성과

학습된 모델을 AutoCar III G의 NVIDIA 기반 Brain Board에 탑재하여 실제 Digital 8 트랙에서 주행 실험을 수행하였다. 직선 구간에서는 비교적 안정적인 주행 성능을 보였으며, 트랙 중앙을 유지하며 큰 진동 없이 주행하는 모습을 확인할 수 있었다.

그러나 급격한 곡선 구간에서는 조향 반응이 충분히 크지 않아 코너를 놓치는 현상이 발생하였다. 이는 출력층에서 sigmoid 활성화 함수를 사용하고, 레이블을 0-1 범위로 강하게 정규화함으로써 조향 값의 해상도가 제한된 것이 주요 원인으로 분석되었다. 특히 서보모터의 물리적 해상도(0.2° 단위)를 고려할 때, 예측 값의 작은 차이가 실제 주행에서는 큰 차이로 이어질 수 있음을 확인하였다.
이 문제를 해결하기 위해, 향후 개선 방향으로는

-출력층 활성화 함수를 tanh로 변경하여 출력 범위를 확장하거나
-조향 값과 속도 값을 분리하여 서로 다른 스케일링 전략을 적용하는 방법
-곡선 구간 데이터를 증강하여 급회전에 대한 학습 비중을 높이는 방법 등을 고려하고 있다.

## 2.2 TAB 의사결정 알고리즘

### 2.2.1 문제 해결 전략

#### (1) 데이터 분석

본 대회에서 제공된 학습용 데이터셋은 총 3개로 구성되어 있으며, 각 데이터셋은 .npy 파일과 이에 대응하는 .json파일로 이루어져 있다. Bandit 알고리즘 설계에 앞서 데이터의 구조적 특성을 명확히 이해하는 것이 필요하다고 판단하여 사전 분석을 수행하였다.

먼저 .npy 파일은 (2000, 2) 형태의 행렬로 구성되어 있으며, 이는 2-armed Bandit 문제를 의미한다. 총 2000회의 시도(Trial) 동안 각 행은 [왼쪽 보상, 오른쪽 보상]의 형태로 주어지며, 각 보상은 성공(1) 또는 실패(0)의 이진 값으로 정의된다. json 파일은 보상 생성 과정의 레벨(Regime) 구조를 포함한다. 전체 2000회의 시도는 약 10개의 레벨로 분할되어 있으며, 각 레벨은 평균적으로 약 200회 내외의 길이를 가진다. 레벨 전환 시점에서는 두 팔의 성공 확률 (p0 ,p1 )이 사전 예고 없이 급격하게 변화한다(예: 0.8:0.2→0.2:0.8). 이러한 특성으로 인해 본 문제는 Non-stationary Bernoulli Bandit 문제로 정의한다.

#### (2) 해결 전략

Bernoulli Bandit 문제를 해결하기 위한 핵심 전제로 다음의 세 가지 전략을 설정하였다.

첫째, 문제의 확률 구조를 적극 활용하는 것이다. 두 구역(A, B)의 확률은 상호 의존적이며, 한쪽 구역에 조난자가 존재할 경우 다른 쪽에 존재하지 않을 확률이 증가한다는 특성을 가진다(p0 +p1 ≈1). 이에 따라 한 번의 선택 결과를 통해 두 팔의 사후 확률을 동시에 갱신하는 전략을 채택함으로써, 관측 정보를 보다 효율적으로 활용하였다.

둘째, 통계적 잔차 기반 급변 감지 기법(CUSUM)의 도입이다. 단순 이동 평균 계열의 방법은 확률 변화에 대한 추종 속도가 느려 환경 변화 감지에 지연(Detection Lag)이 발생하는 한계를 가진다. 이를 보완하기 위해 실제 보상과 예측 확률 간의 차이(잔차)를 누적하여 감시하고, 해당 누적값이 임계값을 초과할 경우 환경이 변화했다고 판단하여 내부 상태를 즉시 초기화(Reset)하는 구조를 설계하였다.

셋째, 앙상블 기법의 활용이다. 단일 파라미터 기반 모델은 특정 노이즈 구간에서 성능이 급격히 저하될

수 있다. 이에 서로 다른 민감도를 가진 다수의 전문가 모델을 병렬로 운용하고, “만약 해당 전문가의 제안을 따랐다면 보상을 획득했을 것인가”를 가상으로 평가하는 Counterfactual Evaluation 방식을 도입하였다. 이를 통해 최근 성과가 가장 우수한 모델을 실시간으로 선택하는 구조를 구현하였다.

### 2.2.2 알고리즘 개발

Armed Bandit 문제 해결에 적합한 확률적 의사결정 기법 가운데, 베르누이 분포 기반 보상 모델에 가장 이론적으로 부합하는 톰슨 샘플링(Thompson Sampling)을 기본 알고리즘으로 채택하였다.

본 문제에서는 A와 B 두 구역 중 하나를 선택하게 되며, 각 구역의 성공 확률은 Beta 분포로 모델링된다. 매 시도마다 각 분포로부터 샘플링된 값을 비교하여 최대값을 갖는 구역을 선택한다. 이 과정에서 관측 데이터가 부족한 구역은 넓은 분포를 유지하여 탐색(Exploration)을 유도하고, 충분한 학습이 이루어진 구역은 분산이 감소하여 활용(Exploitation)이 강화된다. 로봇이 구역 a를 선택하여 보상 r을 획득한 경우, 각 구역의 파라미터는 다음과 같이 갱신된다.

선택 구역 $\alpha_a \leftarrow \alpha_a + r, \beta_a \leftarrow \beta_a + (1-r)$, 반대 구역 $\alpha_{other} \leftarrow \alpha_{other} + (1-r), \beta_{other} \leftarrow \beta_{other} + r$

이는 미러 업데이트(Mirror Update) 기법으로, 실제로 선택하지 않은 구역에 대해서도 간접적인 학습이 가능하도록 한다. 이러한 톰슨 샘플링 구조 위에, 누적합 변화 감지(CUSUM) 기법을 결합하였다. 예측 확률 $\hat{p}_t$과 실제 보상 $r_t$ 간의 편차를 누적하여 다음과 같이 정의한다.

$$g_t^+ = \max(0, g_{t-1}^+ + (r_t - \hat{p}_t) - k)$$

여기서 $k$는 허용 가능한 미세 노이즈(Drift)를 의미하며, 누적합 $g_t$가 임계값 $h$를 초과할 경우 해당 전문가 모델의 $\alpha, \beta$ 파라미터를 1.0으로 초기화한다.

최종적으로, 톰슨 샘플링·미러 업데이트·CUSUM 감지 기법을 결합한 모델을 가상 리더 앙상블(Virtual Leader Ensemble) 구조로 확장하였다. 본 앙상블은 서로 다른 민감도를 가진 4개의 전문가 모델을 병렬로 운용하며, 다양한 트랙 상황(노이즈가 큰 구간과 확률 차이가 명확한 구간)에 유연하게 대응할 수 있도록 설계되었다.

각 시도 이후 마스터 에이전트는 Counterfactual Evaluation을 통해 실제로 선택하지 않은 전문가의 제안이 올바른 선택이었는지를 가상으로 평가한다. 최근 120회의 결과를 기준으로 가장 성과가 우수한 전문가를 실시간 리더로 선출하며, 이때 최근 성과에 높은 가중치를 부여하기 위해 지수 가중 이동 평균(EWMA)을 사용하였다.

$$Score_{i,t} = 0.95 \cdot Score_{i,t-1} + (1 \text{ if } correct \text{ } else \text{ } 0)$$

### 2.2.3 해결하기 어려웠던 점과 그 이유에 대한 고찰

1) CUSUM 민감도 설정의 어려움: 가장 큰 어려움은 CUSUM 임계값 $h$ 설정이었다. 임계값이 지나치게 낮을 경우, 단순한 확률적 변동이나 연속된 실패(0)가 환경 변화로 오인되어 누적된 학습 정보가 불필요하게 초기화되었다. 반대로 임계값이 지나치게 높을 경우, 실제로 환경이 변화했음에도 기존 확률을 고집하여 오답이 누적되는 문제가 발생하였다.

고찰: 환경의 변동성은 제어 불가능한 외생 변수로, 이를 단일 고정 파라미터로 완벽히 대응하는 것은 한계가 있다. 이에 따라 여러 민감도를 가진 모델을 동시에 운용하고, 상황에 따라 가장 성과가 우수한 모델을 선택하는 앙상블 전략이 이 딜레마를 효과적으로 완화할 수 있음을 확인하였다.

2) 마스터 판단 관성(Master Inertia)의 한계: 리더 전문가를 선출하는 과정에서 과거 성적의 비중이 과도할 경우, 새로운 레벨이 시작되어 전문가들이 초기화되었음에도 불구하고 마스터 에이전트가 이전에 성과

가 좋았던 모델을 지속적으로 선택하는 문제가 발생하였다.

해결: 윈도우 기반 평가 방식 대신 감쇠 계수(Decay Factor)를 도입하여, 과거 성과보다 최근 성과에 더 민감하게 반응하도록 평가 로직을 수정하였다. 이를 통해 파일 1과 같이 급격한 확률 변화가 발생하는 구간에서도 점수를 유의미하게 향상시킬 수 있었다.

## 2.3 시스템 통합

강화학습 제출을 진행한 Mirror-CUSUM Ensemble 방식이다. 이 알고리즘은 베이지안 갱신 기반의 Mirror Posterior Update, Double-Mirror Mean Estimation, CUSUM 변화점 탐지 기법을 결합하고, 서로 다른 파라미터를 가진 여러 Expert를 앙상블하여 최고 성능 Expert를 동적으로 선택하는 복잡한 구조를 가지고 있다. 특히 보상 분포가 시간에 따라 변하는 비정상 환경에서 변화점을 자동으로 감지하고 리셋하는 기능은 매우 강력하다. 그러나 본 경기의 LED 보상 분포는 시간에 따라 변하지 않는 고정된 환경이므로, 변화점 탐지 기능은 불필요하다. 또한 4개의 Expert를 관리하고 120개의 가상 보상 이력을 슬라이딩 윈도우로 유지하는 것은 메모리와 연산 측면에서 과도한 부담이며, 매 시행마다 베타 분포 샘플링과 CUSUM 통계량 계산을 수행하는 것은 실시간 제약이 있는 임베디드 환경에서 적합하지 않다.

그렇기에 알고리즘은 고정된 확률 epsilon으로 랜덤 탐색을 수행하고, 나머지 확률로는 현재까지의 평균 보상이 높은 선택지를 착취하는 단순한 구조를 가진다. 연산 복잡도가 O(1)로 매우 낮고 구현이 간단하다는 장점이 있지만, 고정된 epsilon 값은 시간 효율성 측면에서 문제가 된다. 예를 들어 epsilon을 0.3으로 설정한 경우, 초반 몇 번의 시행으로 이미 좌측 구역이 우수하다는 것을 파악했음에도 불구하고, 남은 시행의 30%를 여전히 랜덤 탐색에 사용하게 된다. 5분이라는 제한 시간 내에 최대 40회 정도의 시행만 가능한 상황에서, 후반부의 불필요한 탐색은 총 보상을 직접적으로 감소시키는 요인이 된다.

이러한 분석을 바탕으로 최종 선택한 알고리즘은 Epsilon-Greedy Decay이다. 이 알고리즘은 기본 Epsilon-Greedy의 단순성을 유지하면서도, 시간에 따라 탐색 확률을 점진적으로 감소시켜 시간 효율성을 크게 개선한다. 초기에는 높은 탐색률(예: 0.4)로 양쪽 구역의 정보를 충분히 수집하고, 시행이 진행됨에 따라 탐색률을 감소시켜(예: 0.9배씩 곱하여) 최종적으로 최소 탐색률(예: 0.1)에 도달하도록 한다. 이는 초반에는 정보 수집을 우선시하고, 중반에는 탐색과 착취의 동적 균형을 유지하며, 후반에는 최적 구역에 집중하는 적응적 전략을 자동으로 구현한다.

Epsilon-Greedy Decay의 가장 큰 장점은 연산 복잡도가 매우 낮다는 점이다. 매 시행마다 수행하는 연산은 랜덤 숫자 생성, 단순 비교, 그리고 epsilon 값의 곱셈뿐이며, 이는 모두 O(1) 시간에 완료된다. UCB의 로그 및 제곱근 연산, Mirror-CUSUM의 베타 분포 샘플링이나 CUSUM 통계량 계산과 비교하면 수십 배 이상 빠르다. 이러한 단순성은 실시간 주행 시스템에서 의사결정 지연을 최소화하며, 5분이라는 제한 시간 내에 안정적으로 최대한 많은 시행을 수행할 수 있게 한다.

Epsilon-Greedy Decay는 구현과 디버깅이 매우 간단하다. 조정해야 할 파라미터는 초기 탐색률, 최소 탐색률, 감소율 세 가지뿐이며, 각 파라미터의 의미가 직관적이어서 실험을 통한 튜닝이 용이하다. 복잡한 알고리즘의 경우 수많은 파라미터 조합을 시도해야 하고, 알고리즘 내부 동작을 이해하기 어려워 문제 발생 시 원인 파악이 힘들지만, Epsilon-Greedy Decay는 동작 원리가 명확해 안정적인 운용이 가능하다.

최소 탐색률을 완전히 0으로 만들지 않고 0.1 정도로 유지하는 것도 중요한 설계 선택이다. 이는 환경이 예상과 다르게 변할 가능성에 대비하거나, 초기 샘플의 우연한 편향을 보정하는 역할을 한다. 10%의 탐색은 총 보상에 큰 영향을 주지 않으면서도 로버스트성을 확보하는 안전장치로 작용한다.

결론적으로, Two-Armed Bandit 문제에서 Epsilon-Greedy Decay는 고급 알고리즘들의 불필요한 복잡성을 배제하고, 연산 효율성과 시간 효율성을 모두 확보한 최적의 선택이다. 이론적 완결성보다는 5분 제한 시간 내 실전 성능을 우선시한 공학적 판단이며, 임베디드 환경의 실시간 제약을 고려한 적정 기술의 적용 사례라고 할 수 있다.