DUCKIETOWN

Önvezető autózás a Duckietown környezetben

Csapat:

Tornádó

Team members:

Kohlmann Dániel

Köpeczi-Bócz Ákos Tamás

Széles Katalin

Agenda

Bevezetés

Korábbi megoldások

Rendszerterv

Architektúra, tanítás

Eredmények

Bevezetés

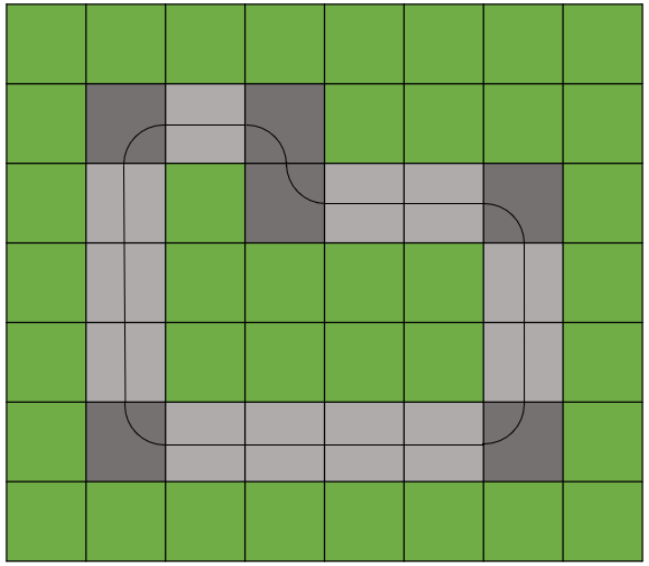
Célunk & Eszközeink

Cél:

- A Reinforcement Learning gyakorlati hátterének részletes megismerése
- Működő önvezető algoritmus tanítása

Duckietown:

- Oktatási és kutatási keretrendszer specifikusan önvezetésre
- Szimulációs környezet az elérhető megvalósításért
- A szimulációs környezetben tanított algoritmus kipróbálható és tovább tanítható a valóságban is



Korábbi megoldások

Korábbi megoldások

1.) Reinforcement Learning keretrendszerek

- Pyqlearning
- KerasRL (OpenAI gym kompatibilis)
- Tensorforce
- Stb.

2.) Tanszéken készített modellek

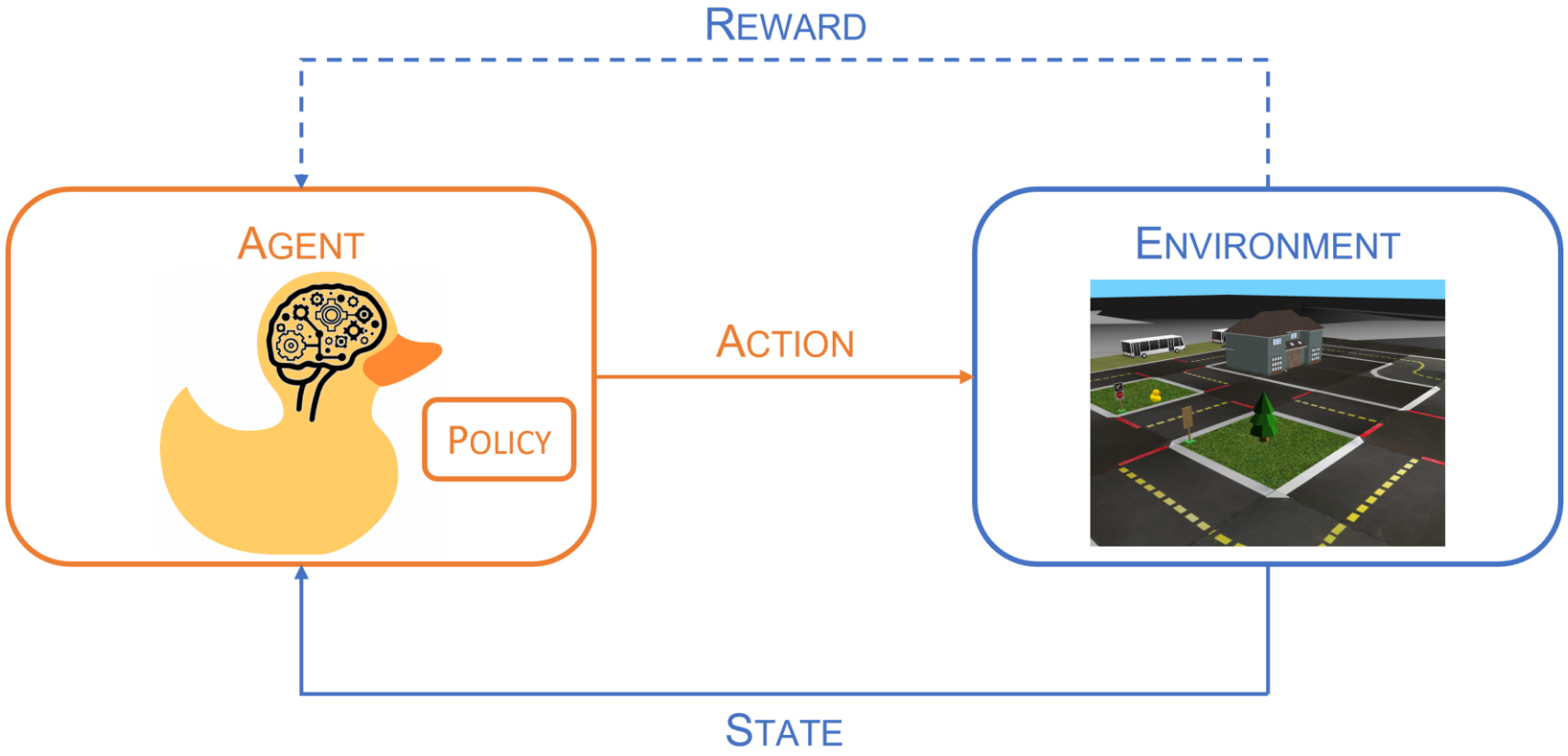
→P. Almási, R. Moni, B. Gyires-Tóth:
Robust Reinforcement Learning-based Autonomous Driving Agent for Simulation and Real World

→A. Kalapos, Cs. Gór, R. Moni, I. Harmati:
Sim-to-real reinforcement learning applied to end-to-end vehicle control

- Kis hálón tanítottak → kis erőforrásigény
- Nem áll rendelkezésünkre nagy erőforrás (GTX 750)
- DQN learning/PPO (Proximal Policy Optimization)
- Tanítás szimulációs környezetben és ezen tanítás átültetése valós környezetben is

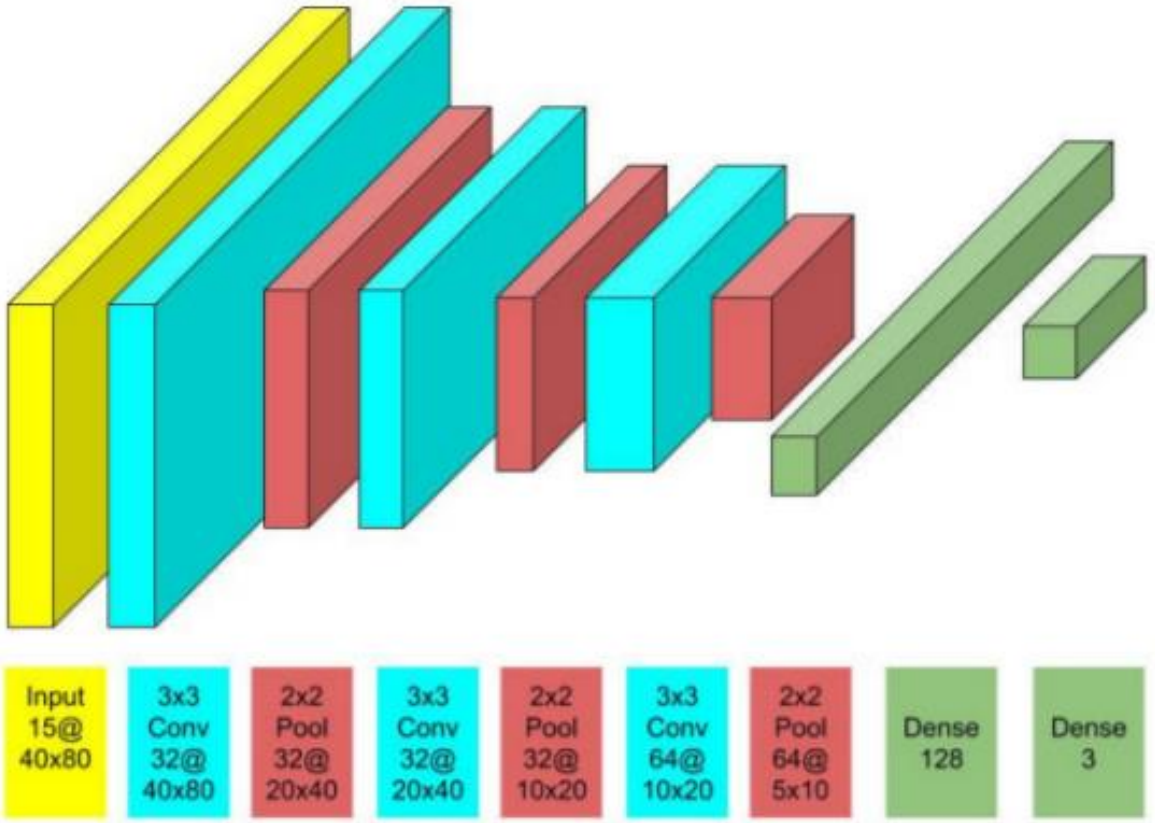
Rendszerterv

RL modell



Architektúra, tanítás

A hálónk



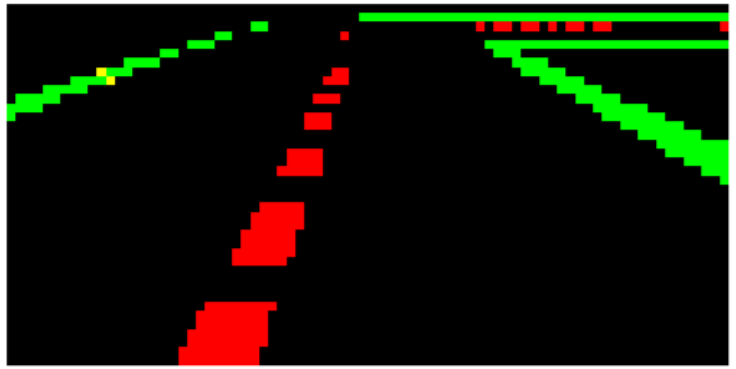
Forrás: P. Almási, R. Moni, B. Gyires-Tóth, et al., "Robust Reinforcement Learning-based Autonomous Driving Agent for Simulation and Real World", in 2020 International Joint Conference on Neural Networks (IJCNN), 2020.

Kezdeti hálóhoz képest:

- Kisebb és konstans kernel méretek (eredetiben csökkenő kernelméretek 8x8-ról indulva)
- Kisebb mélység a 2. és 3. konvolúciós rétegeknél
- Kettővel kevesebb Dense réteg

Adatelőkészítés

- Bemenet: 5 egymásutáni időpillanatból kép
  - A mozgást is érzékelje, ne csak az aktuális pozíciót
- Az ég levágása
- Átkonvertálás: 80x40 pixelméret
- Sárga és fehér sávok szegmentálása
  - Piros és zöld csatornába való mentése
  - A kék csatorna egyelőre üres → potenciális továbbfejlesztési lehetőség
- Értékek normalizálása



Tanítás

- Előzménymemória
- Háló inicializálás
- Minden epizód ciklus esetén:
  - Környezet inicializálása (random seed)
  - Minden körön belül lépésenként:
    - Lépés választása (epsilon-greedy)
    - Lépés megvalósítása
    - Állapotlehívás és reward
    - Adatok mentése az előzménymemóriába
  - Tanítás
    - Batch választása véletlenszerűen az előzménymemóriából
    - Target network segítségével állapotjóslás
    - DQN, Bellman egyenlet
    - Súlyok frissítése
- 3 epizódonként a policy network súlyainak átmásolása a target networkre

Lépésválasztás

- Előzménymemória
- Háló inicializálás
- Minden epizód ciklus esetén:
  - Környezet inicializálása (random seed)
  - Minden körön belül lépésenként:
    - Lépés választása (epsilon-greedy)
    - Lépés megvalósítása
    - Állapotlehívás és reward
    - Adatok mentése az előzménymemóriába
  - Tanítás
    - Batch választása véletlenszerűen az előzménymemóriából
    - Target network segítségével állapotjóslás
    - DQN, Bellman egyenlet
    - Súlyok frissítése
- 3 epizódonként a policy network súlyainak átmásolása a target networkre

- Felfedezés/felhasználás (exploration vs. exploitation)
- Epsilon Greedy metódus
  - $\varepsilon = \max(0.01, 0.01 + (1 - 0.01) \cdot e^{-0.03 \cdot \text{epszám}})$
  - 40 epizód után a döntések 70%-t a háló felhasználásával hozza meg.
  - Teszt üzemmód:
    $\varepsilon = 0.01$

Q-learning

- Előzménymemória
- Háló inicializálás
- Minden epizód ciklus esetén:
  - Környezet inicializálása (random seed)
  - Minden körön belül lépésenként:
    - Lépés választása (epsilon-greedy)
    - Lépés megvalósítása
    - Állapotlehívás és reward
    - Adatok mentése az előzménymemóriába
  - Tanítás
    - Batch választása véletlenszerűen az előzménymemóriából
    - Target network segítségével állapotjóslás
    - DQN, Bellman egyenlet
    - Súlyok frissítése
- 3 epizódonként a policy network súlyainak átmásolása a target networkre

- DQN metódust választottuk
- Cél: minden állapotra a legjobb lépés választása
- Magas dimenziós állapottér
- A várt Q értéket a Bellman egyenlet alapján számoljuk

$$q_*(s,a) = R_{t+1} + \gamma \max(q_*(s',a')$$

Nehézségek

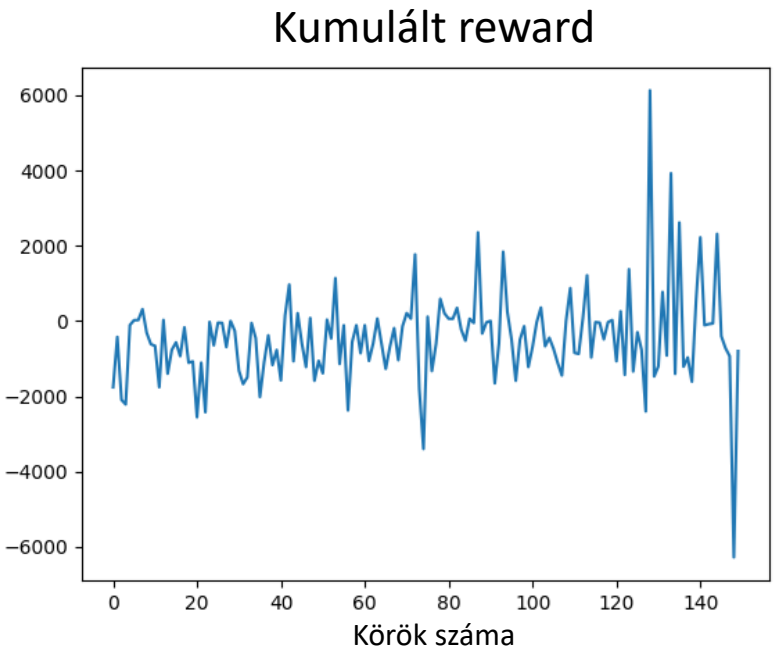
- Szimulációs környezet elindításának nehézségei
  - Megoldás: Linux
- Bonyolult pályán való tanítás kezdetben
  - Megoldás: új, egyszerűbb pálya
- Háló struktúra
  - Megoldás: Forrásaink alapján átstrukturáltuk
- Kép előkészítés
  - Megoldás: CV2 használata
- Q értékek meghatározása
  - Megoldás: Bellman egyenlet használata
- Epsilon-greedy behangolása
  - Megoldás: Manuálisan próbálgatással
- Oszcilláció:
  - Próbálkozásaink:
    - Paraméterek manuális hangolása (batch size, learning rate, mini-batch size)
    - Target network implementálása
    - Reward function állítása (főleg a „collision penalty”-n állítottunk)

Eredmények

Tesztelés

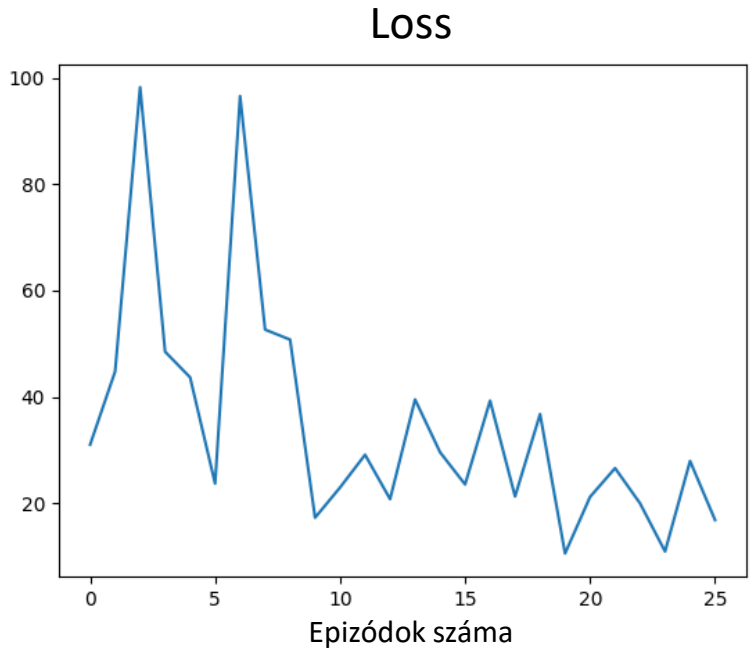
Reward:

- Folyamatában lehet követni
- Kumulált reward egy körre
- Az aktuális rewward function mellett nem a leginformatívabb
  - Megtett út alapján való reward esetén jó metrika



Loss:

- Azt mutatja meg, hogy a háló által választott Q érték és az elvárt Q érték mennyire különbözik
- A mi esetünkben ez volt kezdetben informatívabb



DEMO I.

DEMO II.

Továbbfejlesztési lehetőségek

Fő probléma:

- Oszcilláció (lokális minimum)
  - Balra kanyarodás

Megoldások:

- Reward function

- Hiperparaméter optimalizáció
  - Eddig manuálisan: batch size, learning rate
  - Automatizálni kéne + identifikálni a legbefolyásosabb paramétereket
  - Körönként számolt TOTAL REWARD alapján, amikor már a döntéseink nagy részét a háló alapján hozza a rendszer

- Másik reinforcment learning technikával tesztelés

Összefoglalás

Csapatnév: TORNÁDÓ

Résztvevők: Kohlmann Dániel, Köpeczi-Bócz Ákos Tamás, Széles Katalin

Téma címe: Önvezető autózás a Duckietown környezetben

Elsődleges eredmény: Sávkövető viselkedés szimulációs környezetben (problémák identifikálva)