TheFatBot

Ein Doom-Reinforcement-Learning-Approach mit ViZDOOM

Agenda

1. Einleitung & ViZDoom
2. Szenario erstellen
3. Duel Q-Learning & PPO
4. Unsere Umsetzung & Hyperparameter Tuning
5. Herausforderungen
6. Fazit

Einleitung

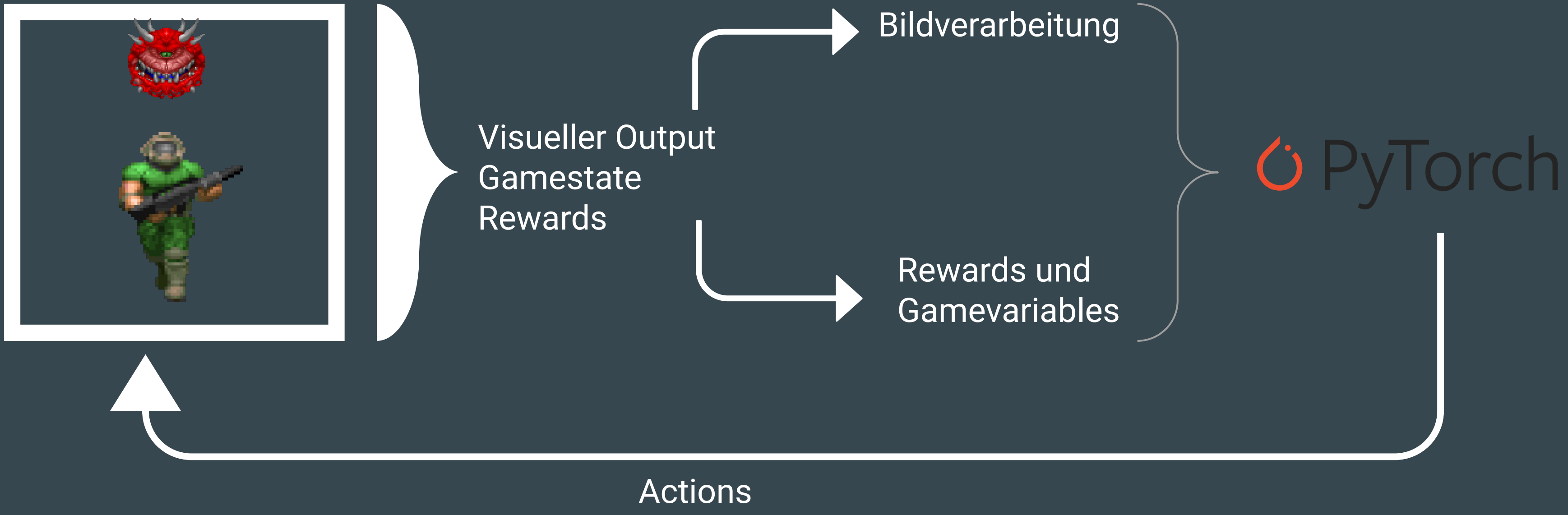
Was ist Doom?

ViZDOOM

Erlaubt uns:

- Zugriff auf Screen Buffer
- Zugriff auf In-Game Variablen
- 4 Control Modes
- Erstellen von eigenen Szenarien
    - Maps
    - dynamisches Spielgeschehen
    - speziell für Reinforcement Learning: Rewards definieren

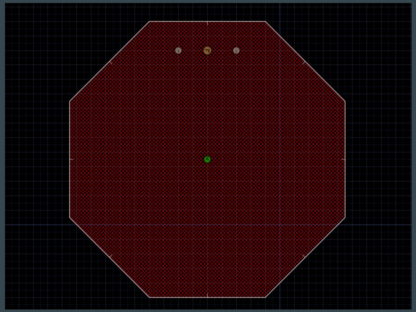
Überblick


Bildverarbeitung
Visueller Output
Gamestate
Rewards
Rewards und
Gamevariables
PyTorch
Actions

Szenarios

Wad Files

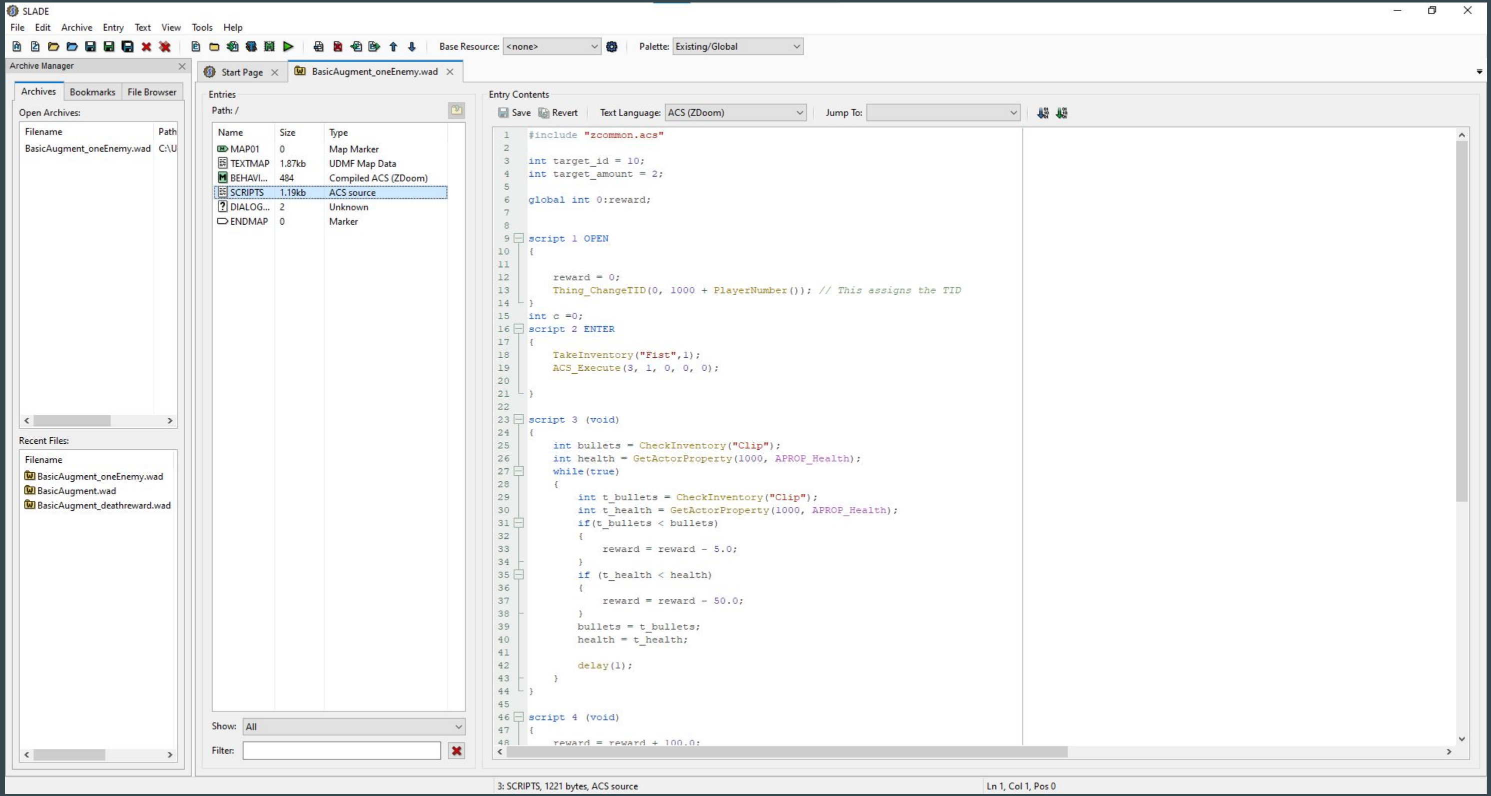
Scripts



Maps

Game Files &
Texturen

Slade

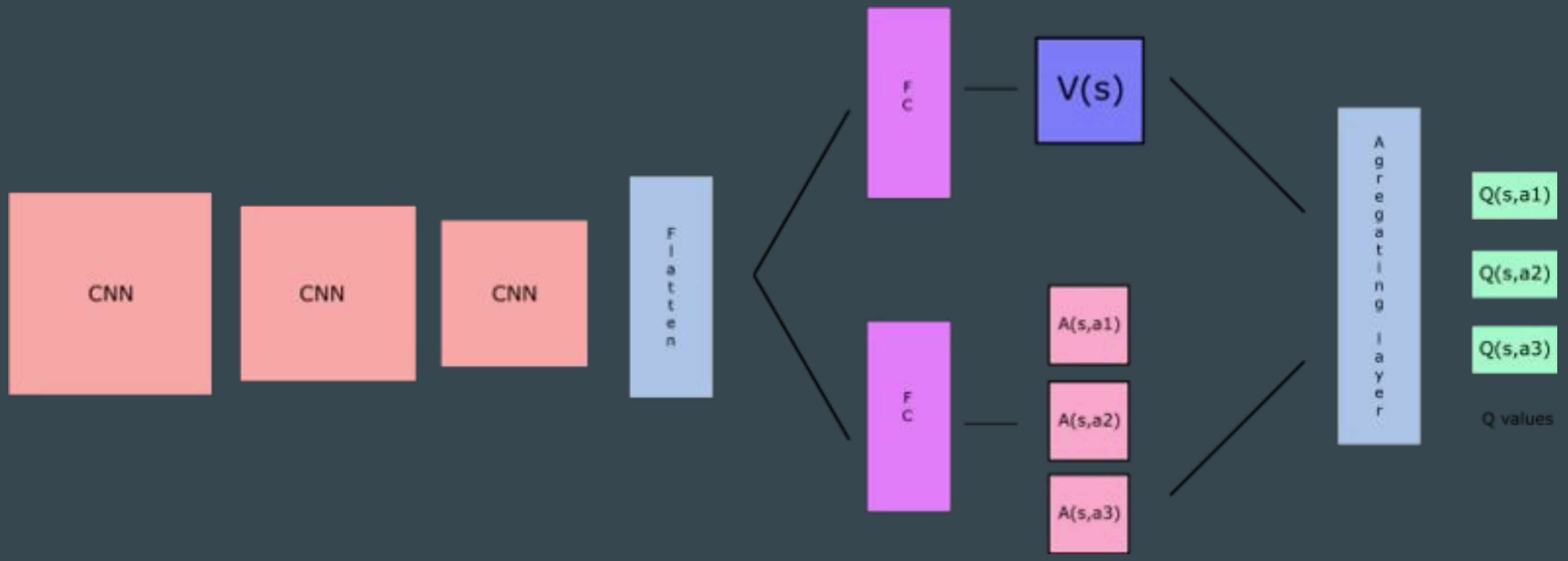

SLADE
File Edit Archive Entry Text View Tools Help
Base Resource: <none>
Palette: Existing/Global
Archive Manager
Archives Bookmarks File Browser
Open Archives:
Filename Path
BasicAugment_oneEnemy.wad C:\U
Recent Files:
Filename
BasicAugment_oneEnemy.wad
BasicAugment.wad
BasicAugment_deathreward.wad
Start Page
BasicAugment_oneEnemy.wad
Entries
Path: /
Name Size Type
MAP01 0 Map Marker
TEXTMAP 1.87kb UDMF Map Data
BEHAVI... 484 Compiled ACS (ZDoom)
SCRIPTS 1.19kb ACS source
DIALOG... 2 Unknown
ENDMAP 0 Marker
Show: All
Filter:
Entry Contents
Save Revert Text Language: ACS (ZDoom) Jump To:
#include "zcommon.acs"
int target_id = 10;
int target_amount = 2;
global int 0:reward;
script 1 OPEN
{
reward = 0;
Thing_ChangeTID(0, 1000 + PlayerNumber()); // This assigns the TID
}
int c =0;
script 2 ENTER
{
TakeInventory("Fist",1);
ACS_Execute(3, 1, 0, 0, 0);
}
script 3 (void)
{
int bullets = CheckInventory("Clip");
int health = GetActorProperty(1000, APROP_Health);
while(true)
{
int t_bullets = CheckInventory("Clip");
int t_health = GetActorProperty(1000, APROP_Health);
if(t_bullets < bullets)
{
reward = reward - 5.0;
}
if (t_health < health)
{
reward = reward - 50.0;
}
bullets = t_bullets;
health = t_health;
delay(1);
}
}
script 4 (void)
{
3: SCRIPTS, 1221 bytes, ACS source
Ln 1, Col 1, Pos 0

Implementierung von Rewards

Aktion → **Skript** → **Reward**

Duel Q-Learning

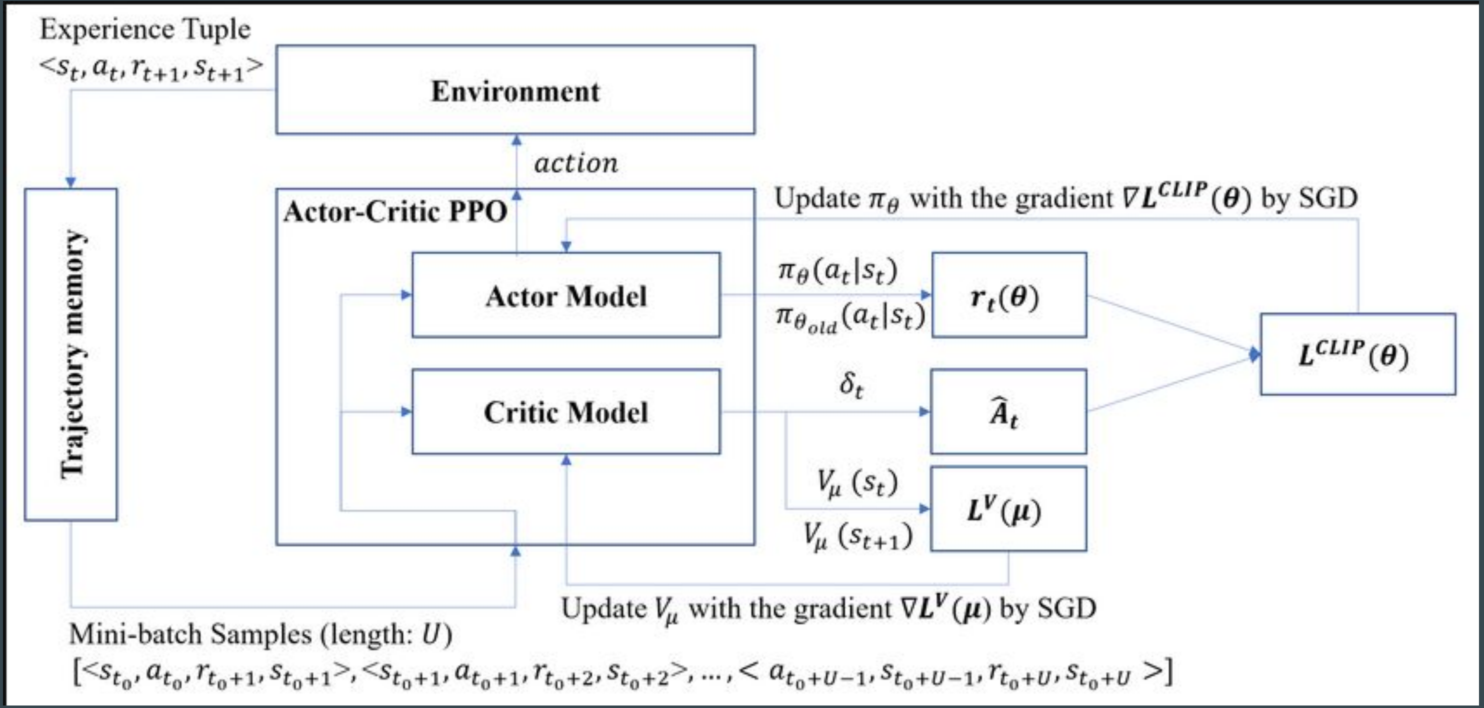
Wie funktioniert Duel Q-Learning?



https://www.freecodecamp.org/news/improvements-in-deep-q-learning-dueling-double-dqn-prioritized-experience-replay-and-fixed-58b130cc5682/

Proximal Policy Optimization

Wie funktioniert PPO?



Unsere Umsetzung

Die Ausgangslage

Erste Schritte in ViZDoom

Basic.wad

My Way Home.wad

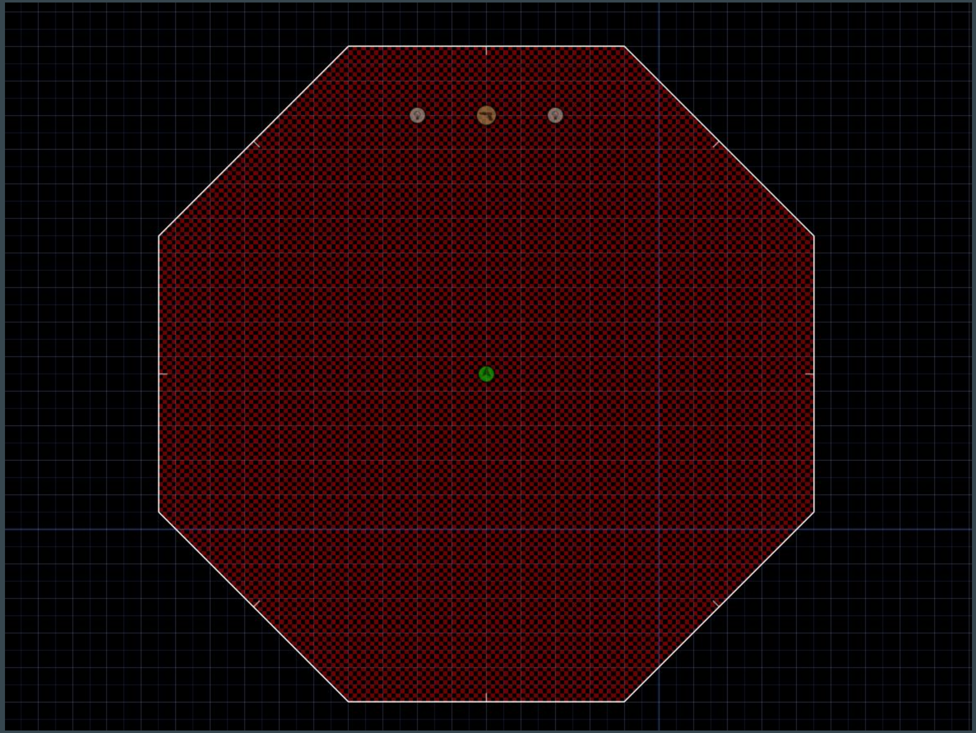
Challenges from the get go

Sparse Rewards

Balance Rewards

Sensitivity to Parameters

Unsere Map



Unser Reward script

Positive Rewards:

- Gegner Treffer
- Aufsammeln der Waffe

Negative Rewards

- Schießen
- Schaden nehmen
- Game Over

Experimentelle Rewards:

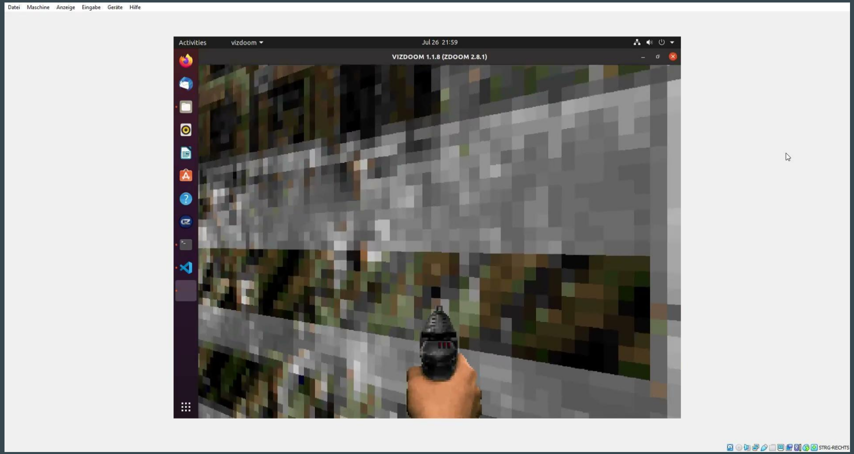
- Negativer Living Reward
- Rewards auf Positionsbasis

Hyper Parameter Tuning und Vergleich

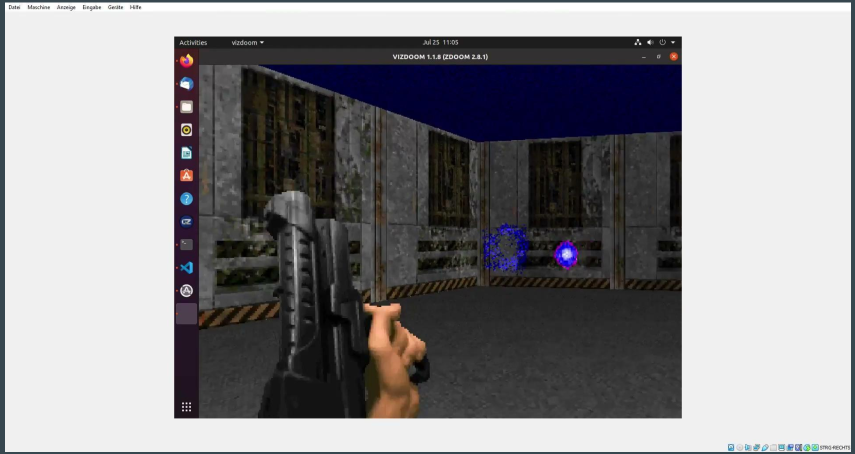
Hyper Parameter

1. Anzahl der Epochen
2. Learning Rate
3. Discount Factor
4. Learning Steps per Epoch
5. Replay Memory Size
6. Batch Size

Hyperparameter Tuning

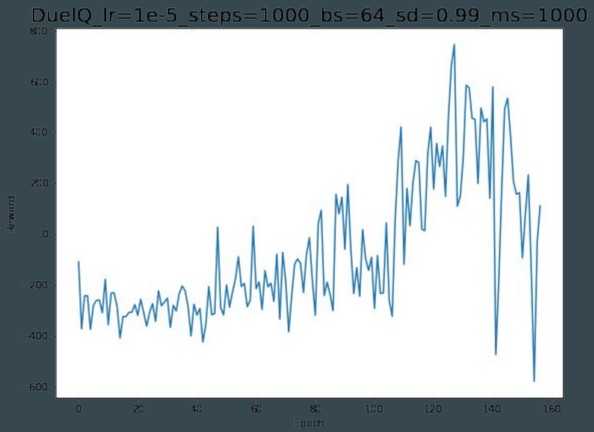


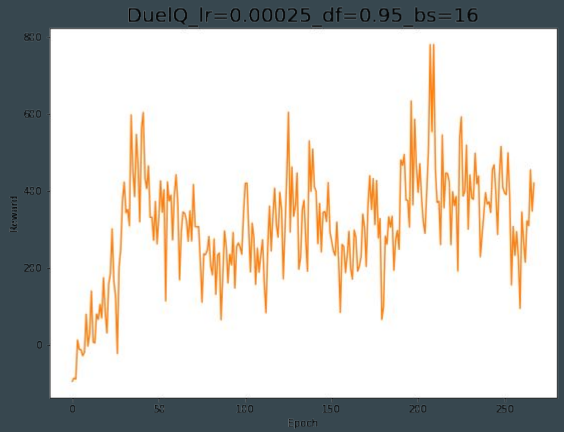
e=391 lr=0,0005

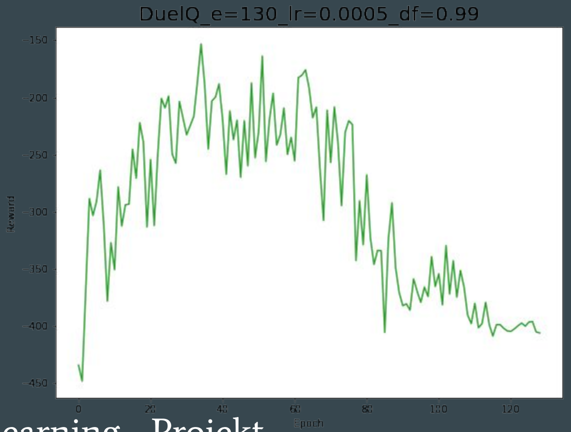


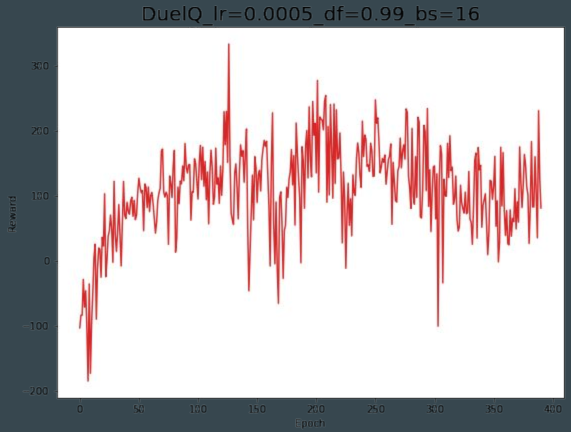
e=412 lr=0,0002

Bewertung


DuelQ_lr=1e-5_steps=1000_bs=64_sd=0.99_ms=1000
Reward
Epoch


DuelQ_lr=0.00025_df=0.95_bs=16
Reward
Epoch


DuelQ_e=130_lr=0.0005_df=0.99
Reward
Epoch


DuelQ_lr=0.0005_df=0.99_bs=16
Reward
Epoch

Gescheiterte Umsetzungen

Proximal Policy Optimization

Imitation Learning

Fazit

Fazit

Eher schwache Performance
der Modelle

- Zu wenig Rechenleistung
    - Begrenzung durch VMs
- Probleme bei Reward Funktion
- Zeitlich zu aufwändig
- Umsetzung von weiteren RL Ansätzen an Bugs gescheitert

Danke für eure Aufmerksamkeit!

Quelle

- Federated Reinforcement Learning for Training Control Policies on Multiple IoT Devices - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-actor-critic-proximal-policy-optimization-Actor-Critic-PPO-algorithm-process_fig3_339651408 [accessed 27 Jul, 2021]
- Reinforcement Trained Basic Example: https://www.youtube.com/watch?v=fKHw3wmT_uA
- Reinforcement Trained My Way Home Example: https://www.youtube.com/watch?v=15yZubaTLvw
- Vizdoom Proposal: https://arxiv.org/abs/1605.02097 [accessed 26 Jul, 2021]
- Proximal Policy Optimization Paper: https://arxiv.org/abs/1707.06347 [accessed 26 Jul, 2021]
- Duel Q Learning Proposal: https://arxiv.org/abs/1511.06581 [accessed 26 Jul, 2021]