Connect 4

Team HiddenName

Jacqueline Wegert, Jonathan Merkel, Frederik Dangel

Übersicht

- Hintergrund und Ansatz
- Verwendete Methoden und Training
- Evaluation und Ausblick

Hintergrund - Kaggle Challenge “Connect X”

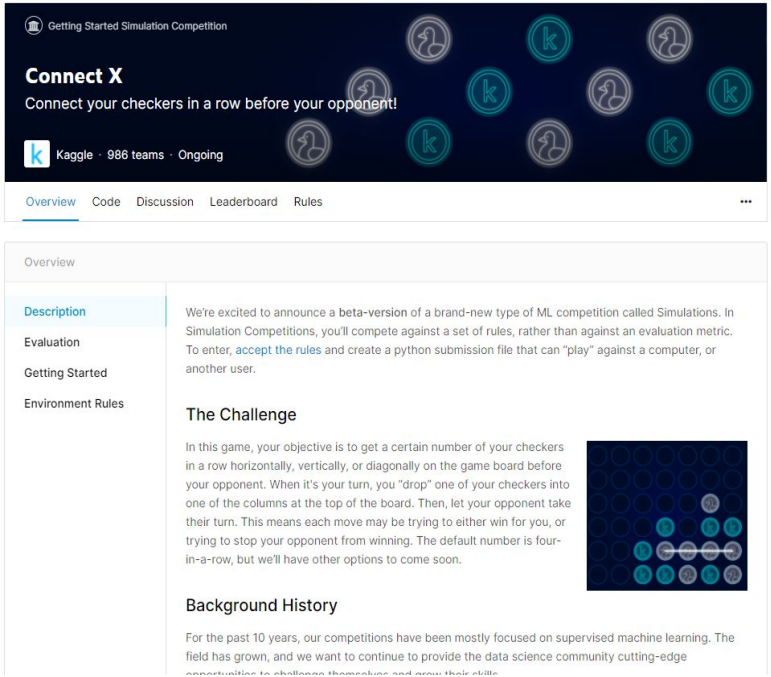
- Möglichkeit zur Konfiguration des Spielfelds
- Stellt Environment zur Verfügung
- Default Agents verfügbar

```
from kaggle_environments import make
env = make("connectx", {"rows": 10, "columns": 8, "inarow": 5})

def agent(observation, configuration):
  print(observation) # {board: [...], mark: 1}
  print(configuration) # {rows: 10, columns: 8, inarow: 5}
  return 3 # Action: always place a mark in the 3rd column.

# Run an episode using the agent above vs the default random agent.
env.run([agent, "random"])

# Print schemas from the specification.
print(env.specification.observation)
print(env.specification.configuration)
print(env.specification.action)
```

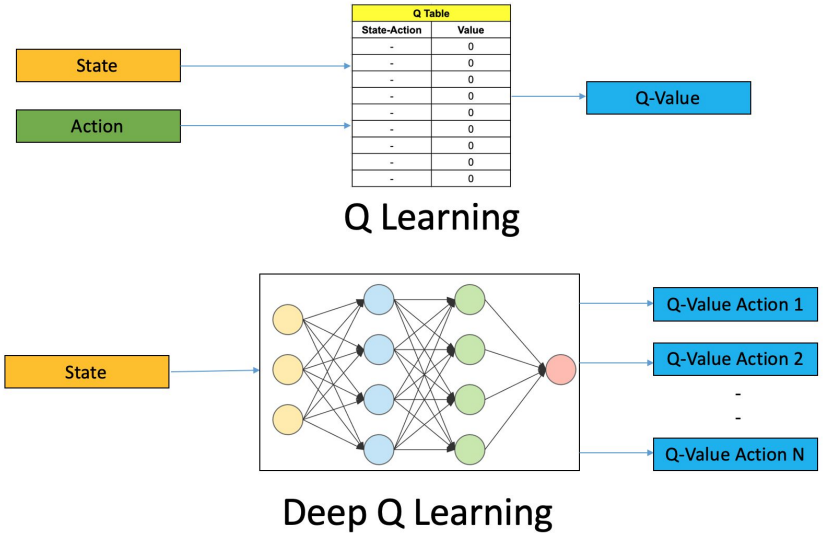


https://pypi.org/project/kaggle-environments/, https://www.kaggle.com/c/connectx/overview

Ansatz

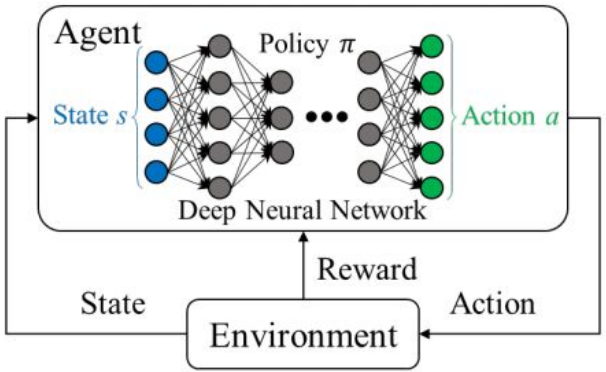
- Anzahl möglicher Kombinationen zu groß für Q-Table
- Umsetzung mit Deep Q Network

n	a(n)
0	1
1	7
2	49
3	238
4	1120
5	4263
10	1662623
15	176541259
20	6746155945
25	97266114959
30	410378505447
35	370947887723
40	22695896495
41	7811825938
42	1459332899



https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/

Ansatz DQN

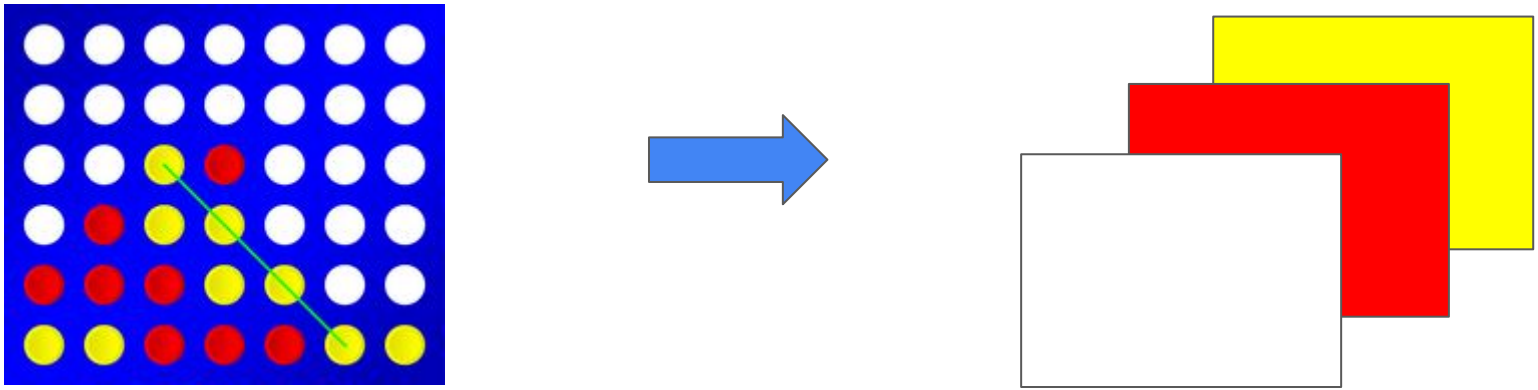


- State space s: mögliche Situationen, die im Spiel auftreten können
- Action space a: erlaubte Spalten, in die ein Spielstein eingeworfen werden kann
- Policy π: Wahrscheinlichkeitsverteiung über die Spielzustände und die Spalten
- Rewards:
  - 1 für Gewinnen
  - -1 für Verlieren
  - 0 für Unentschieden

https://ieeexplore.ieee.org/document/8742652

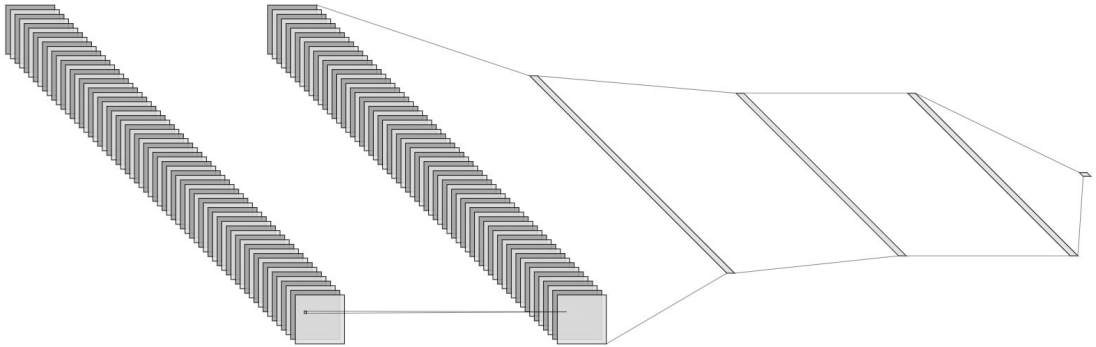
Data Preparation

1Hot Encoding des Spielbretts



Netzarchitektur

- CNN (besser dank 1Hot)
- 2 Conv2D mit 64 Features & Kernel Size 3
- ReLU Aktivierungsfunktion
- Flatten, Fully Connected
- Output: 7 Neuronen



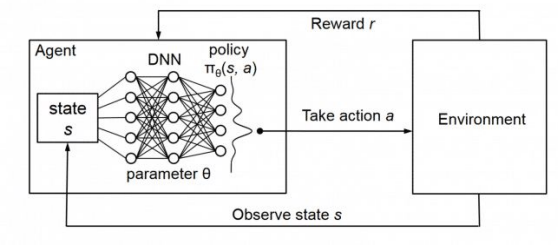
Anpassungen des Rewards

- Schnell ersichtlich: Reward besitzt starken Einfluss
- Invalid Move → irreguläre Züge Lernen
- Step → Verlängerung der Spieldauer

Result	Reward	Reward with rows
Win	10	20
Loss	-10	-20
Invalid Move	-20	-40
Draw	1	1
Step	1/42	1/42 + 3*(no. three in a row)

Training

- Gegner: Random, (negamax, eigens trainierter Agent)
- Experience Replay Buffer (Buffer_size: 20.000)
- Batch_size: 32
- Netze synchronized alle 50 Episoden
- ε-Greedy Policy mit decay 0.9 → 0.1 in 1500 Episoden
- alle 1000 Episoden: tausche Spieler 1→ 2 & umgekehrt
- Optimizer Adam(lr=0.001) mit MSE loss
- jeder Trainings run umfasst 20.000 Episoden



Evaluation

- Gegner: random & negamax
- Parameter mit größtem Einfluss: reward und Diskontierungsfaktor
- Training verbessert sich, wenn 3er-Kombinationen Reward bringt
- Diskontierungsfaktor [0.4, …, 1]

Gewinnanteil in der Evaluation

Default reward

Discount factor	Random	Negamax
0.4	97%	1%
0.5	96%	2%
0.6	92%	3%
0.7	94%	7%
0.8	95%	2%
0.9	86%	3%
1.0	86%	0%

Reward für 3er Kombination

Discount factor	Random	Negamax
0.4	98%	7%
0.5	97%	7%
0.6	98%	4%
0.7	96%	9%
0.8	91%	8%
0.9	78%	6%
1.0	58%	2%

Irreguläre Züge erlauben

Evaluation Configuration	Random	Negamax
Default Reward Disallow invalid actions	69%	6%
Additional Reward Disallow invalid actions	74%	5%
Default Reward Allow invalid actions	59%	0%
Additional Reward Allow invalid actions	65%	0%

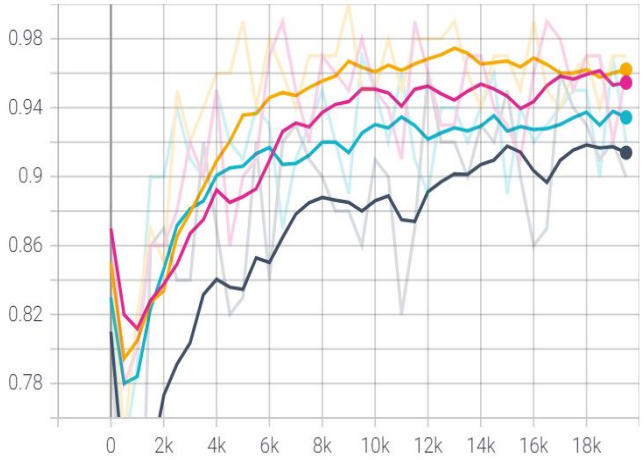
Evaluation Configuration	Invalid Actions Random	Invalid Actions Negamax
Default Reward Allow invalid actions	11	1
Additional Reward Allow invalid actions	9	6

Training mit zu wenig neuen Episoden

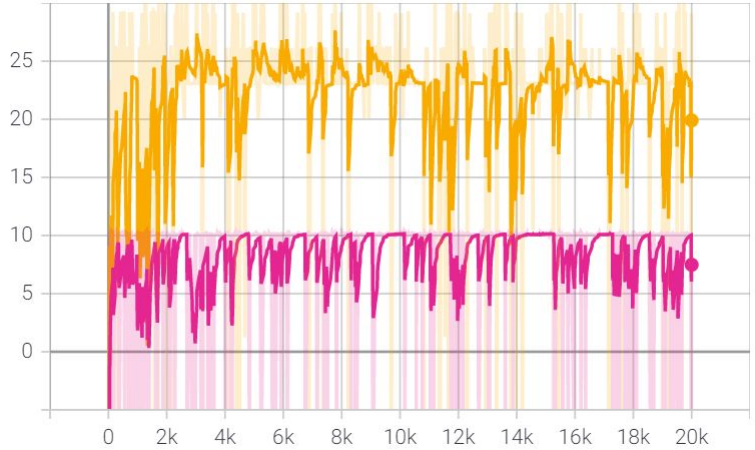
Configuration	Random	Negamax
Discount: 0.7 Default Reward	93%	3%
Discount: 0.7 Streak Reward	90%	4%

Entwicklung während dem Training

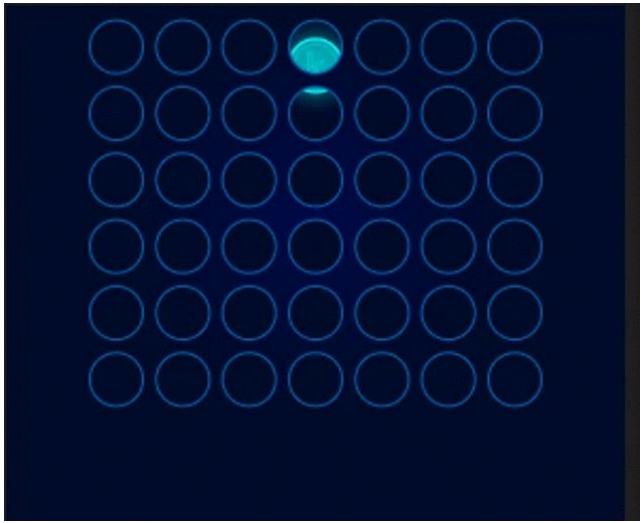
Gewinnwahrscheinlichkeit



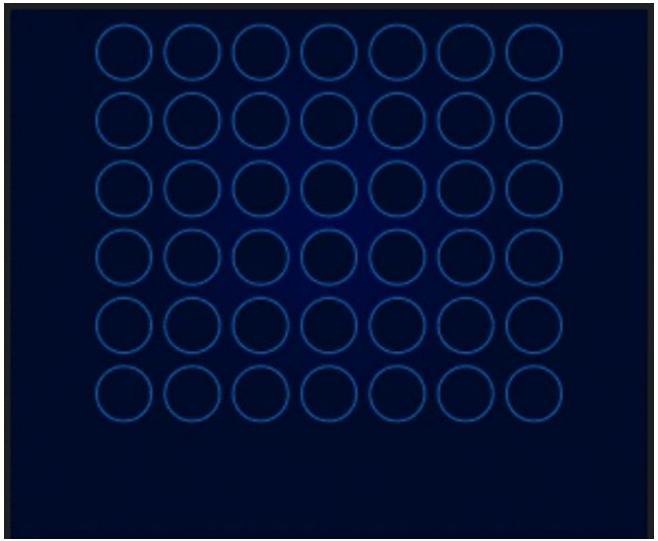
Batch reward



Aufzeichnung gegen den Random Bot



Aufzeichnung gegen den Negamax Bot



Key Takeaways

- Starke Abhängigkeit zum Reward System & Diskontierungsfaktor, geringe zum CNN
- CNN mit 1-Hot besser als Fully Connected Network
- Netz lernt nicht selbst, welche Aktion gültig ist
- Nach etwa 10.000 Episoden hinreichend trainiert

Ausblick & Weiterentwicklungsmöglichkeiten

- Training gegen Suchalgorithmus oder gegen sich selber
- Anpassung des Reward-Systems um Entwicklung des Agents zu lenken
- Bootstrapped DQN → effizientere Exploration [3]
- Wichtige State Transitions bevorzugt aus dem Replay Buffer in Trainings Batch für effizienteres Lernen [5]
- Double DQN [6]
- Noise für Exploration um Noise in den Parametern des Agenten ergänzen [7]

Verwandte Arbeiten

[1] Stefan Edelkamp and Peter Kissmann. “Symbolic Classification of General Two-Player Games”. In:KI 2008:Advances in Artificial Intelligence. Ed. by Andreas R. Dengel et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 185–192. ISBN: 978-3-540-85845-4.

[2] Volodymyr Mnih et al. Playing Atari with Deep Reinforcement Learning. 2013. arXiv: 1312.5602[cs.LG].

[3] Ian Osband et al. “Deep Exploration via Bootstrapped DQN”. In: CoRR abs/1602.04621 (2016). arXiv: 1602 .04621.URL: http://arxiv.org/abs/1602.04621.

[4] Will Dabney et al. “Implicit Quantile Networks for Distributional Reinforcement Learning”. In:Proceedings of the 35th International Conference on Machine Learning. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 1096–1105.URL: http://proceedings.mlr.press/v80/dabney18a.html.

[5] Tom Schaul et al.Prioritized Experience Replay. citearxiv:1511.05952Comment: Published at ICLR 2016.2015.URL: http://arxiv.org/abs/1511.05952.

[6] Hado van Hasselt, Arthur Guez, and David Silver. “Deep Reinforcement Learning with Double Q-learning”. In: CoRR abs/1509.06461 (2015). arXiv: 1509.06461. URL:http://arxiv.org/abs/1509.06461.

[7] Matthias Plappert et al. “Parameter Space Noise for Exploration”. In: CoRR abs/1706.01905 (2017). arXiv: 1706.01905. URL: http://arxiv.org/abs/1706.01905

[8] John Tromp. A212693 Number of legal 7 X 6 ConnectFour positions after n plies. URL: https : / /oeis .org /A212693.

[9]Deep Reinforcement Learning. https://www.kaggle.com/alexisbcook / deep - reinforcement - learning. Accessed:2021-06-27.