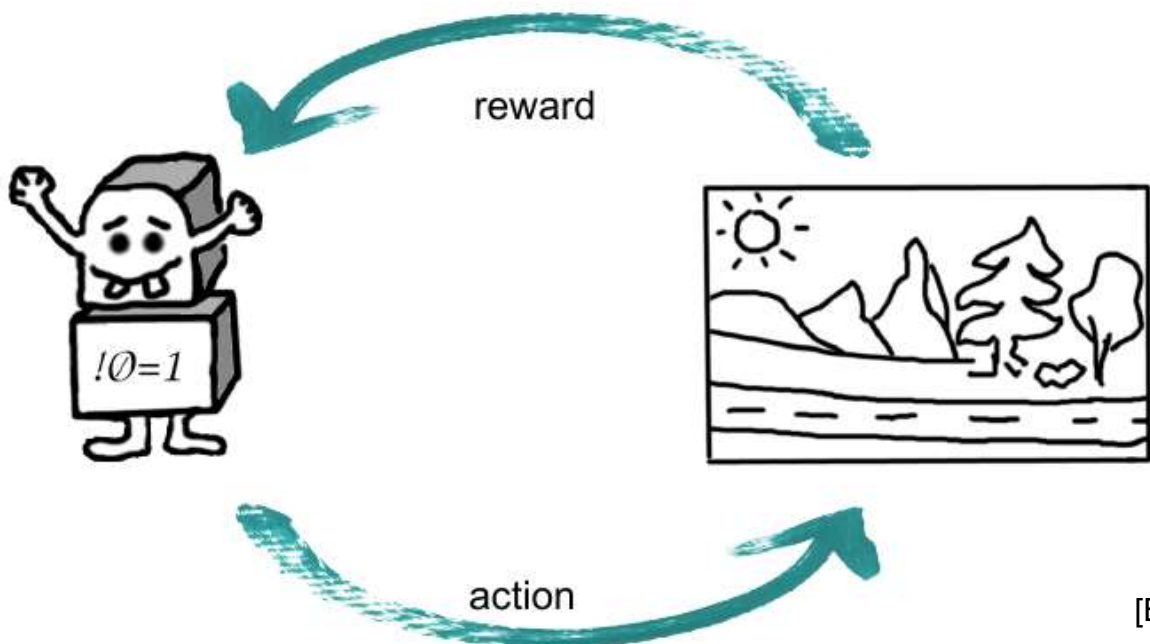




KI programmieren im Informatikunterricht Teil 4 Reinforcement Learning



[B1]

Abb. [B1] „AAEOR“, Alexander Schindler, Lizenz [CC BY-SA4.0](https://creativecommons.org/licenses/by-sa/4.0/), Lernaufgabe "KI im Informatikunterricht 4





Inhaltsverzeichnis

KI programmieren im Informatikunterricht Teil 4.....	1
A Überblick.....	3
A 1 Zusammenfassung:.....	4
A2 Stundenübersicht	5
A3 Themeneinstieg und theoretische Grundlagen.....	6
B Lernaufgabe	7
B1 Unterrichtsbegleitende Präsentation	7
B2 Arbeitsblätter	20
C Hinweise für die Lehrkraft.....	23
C1 Musterlösung	23
C2 Hinweise zur Umsetzung im Unterricht	25
C3 Didaktischer Kommentar.....	26
C4 Bezüge zum Rahmenlehrplan Informatik	26
C5 Bezüge zum Basiscurriculum Medienbildung.....	28
C6 Bezüge zu übergreifenden Themen	28
C6 Bezüge zu anderen Fächern:.....	29
D Anhang	30
D3 Quellen / Lektüreliste zum Weiterlesen.....	30
D4 Bildnachweise / Abbildungen	30



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie - Fachset Informatik
KI Programmieren im Unterricht Teil 4:
Reinforcement Learning
Alexander Schindler



Senatsverwaltung
für Bildung, Jugend
und Familie

BERLIN



A Überblick

Unterrichtsfach	Informatik
Jahrgangsstufe(n)	10-13
Niveaustufe	verschiedene (siehe C4 Bezüge zum Rahmenlehrplan Informatik)
Zeitraumen	Zwei Doppelstunden
Thema	Künstliche Intelligenz
Themenfeld(er)	
Kontext	
Zusammenfassung des Inhalts	<p>Wer neuronale Netze einmal selbst trainiert hat, kann deren Risiken, Probleme und Chancen und damit auch mögliche gesellschaftliche Entwicklungen besser einschätzen.</p> <p>In dieser Lernaufgabe lernen die Schüler:innen den Ablauf beim Reinforcement Learning RL. Im Anschluss ist es den Schüler:innen möglich, eigene Projekte durchzuführen.</p> <p>Diese Lernaufgabe setzt die Grundlagen der ersten Lernaufgabe: „KI programmieren im Informatikunterricht Teil I: Einführung“ voraus. In diesem Sinne: <i>All hands on code!</i></p>
Didaktischer Kommentar	<p>Ziel der Lernaufgabe ist es, SuS einen Einstieg in die Programmierung von reinforcement learning RL zu ermöglichen.</p> <p>Der Code ist so einfach wie möglich gehalten. Die Aufgabe ist skalierbar, d.h. sie kann um eigene Ideen (z. B. <i>environments</i> auf www.gymlibrary.dev) erweitert werden und sie kann und soll somit Basis für eigene Projekte sein.</p> <p>Der Schluss ist ein Ausblick auf mögliche Auswirkungen von RL auf die Lebenswelt der SuS und somit ist die Lernaufgabe auch im Bereich Informatik und Gesellschaft zu verorten.</p>
Schlüsselwörter	python, TensorFlow, gymnasium, gym, keras-rl2, KI, Künstliche Intelligenz, AI, artificial intelligence, IDE, RL, reinforcement learning, Cart Pole, Informatik und Gesellschaft, Software, programmieren, Projekt

Material für den Einsatz dieser Lernaufgabe

Anzahl	Name des Materials
	<ul style="list-style-type: none"> • Schüler und Schülerinnen benötigen eine entsprechend leistungsfähige Python IDE, es empfiehlt sich Spyder. • Python (derzeit Januar 2024 Version 3.9.15, 64 Bit) auch andere Versionen können funktionieren. Eine zu aktuelle Pythonversion ist aber oft hinderlich. • Da die jeweils aktuelle TensorFlowbibliothek in einer nicht aktuellen Pythonversion entwickelt wird, hilft es bei (unklaren) Problemen beim Ausführen eine eher ältere Pythonversion zu verwenden. • Alle Bibliotheken lassen sich über <code>pip install <Bibliothek></code> installieren. Am besten unten stehende Reihenfolge einhalten. Folgende Bibliotheken müssen in der angegebenen Version installiert sein: <ol style="list-style-type: none"> 1. <code>tensorflow == 2.5.0</code> 2. <code>gym == 0.17.2</code> 3. <code>keras-rl2 == 1.0.5</code> 4. <code>gymnasium == 0.26.2</code> 5. <code>gymnasium[classic_control]</code>

A 1 Zusammenfassung:

Intention der Lernaufgabe

- Kennnlernen der Grundlagen von *reinforcement learning*
- Kennnlernen Erstellung / Programmierung eines eigenen Projektes mithilfe der Bibliotheken `gymnasium` und `keras-rl2`
- Erkennen von Chancen und Risiken, möglichen gesellschaftlichen Auswirkungen.

Hinweise:

Das Zusammenspiel der Bibliotheken TensorFlow, gym, keras-rl2 und Python läuft nicht immer problemlos. Bei unklaren Schwierigkeiten hilft es eine eher ältere Pythonversion mit einer eher neueren TensorFlow-Version zu benutzen. Diese Aufgaben wurden mit folgender Konfiguration getestet:



CC BY SA 4.0
 Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie - Fachset Informatik
 KI programmieren im Unterricht Teil 4:
 Reinforcement Learning
 Alexander Schindler



Senatsverwaltung
 für Bildung, Jugend
 und Familie

BERLIN





Siehe auch Installationshinweise unter

- Python 3.9.15, 64-Bit
- tensorflow == 2.5.0
- gym == 0.17.2
- keras-rl2 == 1.0.5
- gymnasium == 0.26.2
- gymnasium[classic_control])

A2 Stundenübersicht

1. Doppelstunde:

- Hinführung zum Thema
- Grundlagen der Erstellung einer policy mit klassischer Programmierung (if...else...)

Zeit	Phase	Beschreibung	Methode/Medien/Anmerkung
5 Min.	Einstieg	Ein Kind schaukelt (<i>action</i>) auf einem Spielplatz (<i>environment</i>) und fühlt sich glücklich (<i>reward</i>). Geht es morgen wieder schaukeln? Impuls zur Bedeutung und Problematik von selbst lernenden Sysmten wie z. B. Robotik, Spiele	Präsentation
10 Min.	Vortrag / LSG	<i>Lehrevortrag zu Grundlagen des reinforcement learnings</i>	Präsentation
10 Min.		Besprechung Quellcode	
55 Min.	Erarbeitung	Aufgaben 1-3	
10 Min.		Diskussion der Ergebnisse	

LSG: Lehrer-Schüler-Gespräch



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie - Fachset Informatik
KI programmieren im Unterricht Teil 4:
Reinforcement Learning
Alexander Schindler



Senatsverwaltung
für Bildung, Jugend
und Familie

BERLIN



2. Doppelstunde

Zeit	Phase	Beschreibung	Methode/Medien/Anmerkung
10 Min.		Besprechung Quellcode	
60 Min.	Erarbeitung	Aufgaben 4-6	
10 Min.		Diskussion der Ergebnisse	
10 Min.		Abschlussdiskussion	Folie „4. Agenten können“
-- Min.	Erarbeitung	Planung / Erstellung eines eigenen Projektes mit Anwendung/ Veränderung des bestehenden Modells	Puffer

LSG: Lehrer-Schüler-Gespräch

A3 Themeneinstieg und theoretische Grundlagen

Theoretische Grundlagen

- Der Verwendete Agent (`CEMAgent`) ermöglicht den einfachsten Code zur Lösung des Cart Pole Problems mit Hilfe von RL. Er basiert auf CEM.
- *CEM Cross Entropy Method*: Aus biologischer Sicht handelt es sich um einen evolutionären Algorithmus. Aus einer Population werden einige Individuen entnommen, und nur die besten bestimmen die Merkmale künftiger Generationen.
- Das Ausschalten der Visualisierung (`visualize=True`) beschleunigt das Training (im Test um das 50-fache).

Lernaufgabe: B Lernaufgabe

B Lernaufgabe



B1 Unterrichtsbegleitende Präsentation



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie - Fachset Informatik
KI programmieren im Unterricht Teil 4:
Reinforcement Learning
Alexander Schindler



Senatsverwaltung
für Bildung, Jugend
und Familie

BERLIN



1. Reinforcement Learning RL



Schaukeln macht Spaß.
Gehen wir morgen wieder
auf den Spielplatz?

[B2]

Abb. [B2]: „We visited Zoom Torino : my daughter Chloé enjoys the zoo playground ! (Turin / Italy - 08/2015) #zoo #zoomtorino #bioparc #playground #Torino #Turin #child #kids #enfant #travelwithkids #familytravel #familytravel #familytrip #cute #happy #smile #fun We“, by leszed.com, Lizenz [CC BY-SA 2.0](https://creativecommons.org/licenses/by-sa/2.0/), [flickr](https://www.flickr.com/photos/leszed/)



CC BY SA 4.0
Ausgenommen sind einzeln ge-kennzeichnete Inhalte/Elemente,
siehe Quellen- und Lizenzhinweise
am Ende des Dokuments.



1. RL - learning cycle

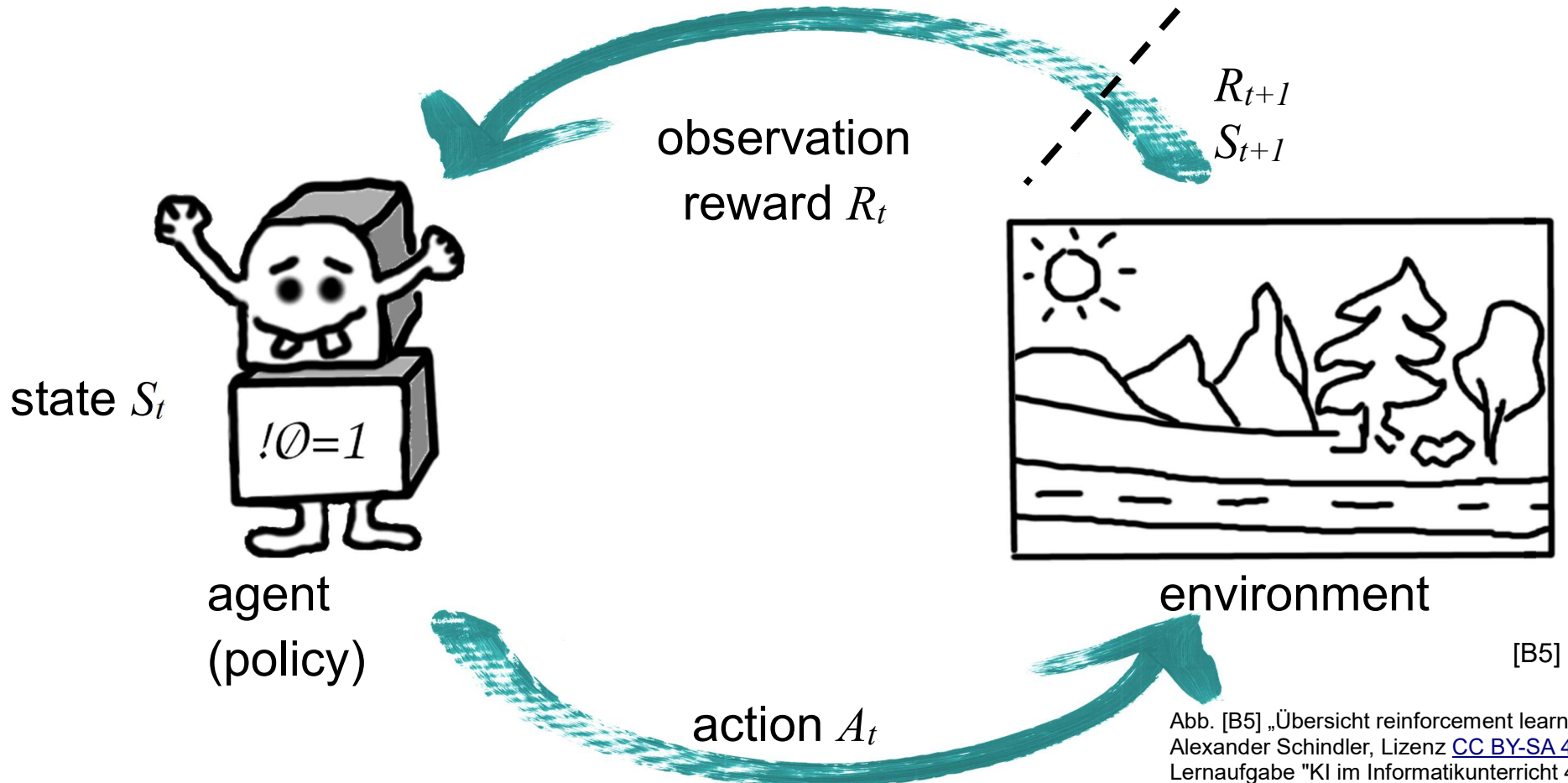


Abb. [B5] „Übersicht reinforcement learning“, Alexander Schindler, Lizenz [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/), Lernaufgabe "KI im Informatikunterricht 4"

1. AREA



A – Action
R – Reward
E – Environment
A – Agent

Ziel: Iterative Verbesserung der *policy* (=Strategie)



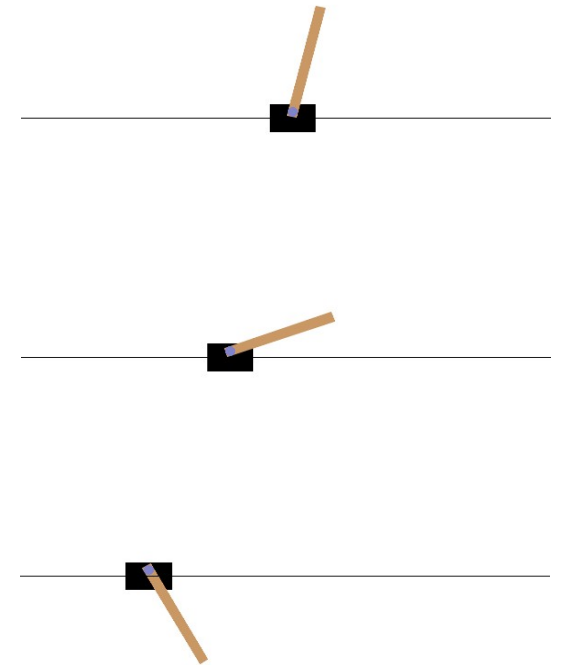
CC BY SA 4.0
Ausgenommen sind einzeln ge-kennzeichnete Inhalte/Elemente,
siehe Quellen- und Lizenzhinweise
am Ende des Dokuments.



1. Reinforcement Learning



- klassisches Problem: CartPol
- Umgebung: `gymnasium / gym`
- Framework: `keras-rl2`
- Neuronales Netzwerk: `TensorFlow`



[B3]

Abb. [B3] „Cart Pole – Screenshots 1“, Alexander Schindler, Lizenz [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/), Lernaufgabe "KI im Informatikunterricht 4"



CC BY SA 4.0
Ausgenommen sind einzeln ge-kennzeichnete Inhalte/Elemente,
siehe Quellen- und Lizenzhinweise
am Ende des Dokuments.



2. CartPole



- Pendel senkrecht balancieren
- *action space*: Schlitten bewegen
- *observation space*: Pendel und Schlitten
- erfolgreich wenn mehr als 500 Episoden balanciert wurden
- Interaktion mit *environment* (Umgebung)
- lernen durch *reward* (Belohnung)

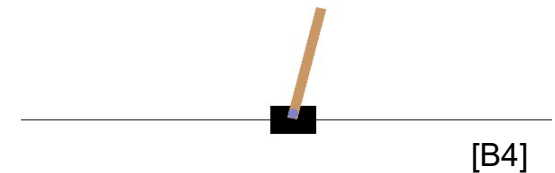


Abb. [B4] „Cart Pole – Screenshots 2“, Alexander Schindler, Lizenz [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/), Lernaufgabe "KI im Informatikunterricht 4"



CC BY SA 4.0
Ausgenommen sind einzeln ge-kennzeichnete Inhalte/Elemente,
siehe Quellen- und Lizenzhinweise
am Ende des Dokuments.



2. CartPole - Abbruchkriterien



- terminated
 - Schlittenposition: $\text{observation}[0] > \pm 2.4$
 - Pendelwinkel: $\text{observation}[2] > \pm 12^\circ$
- truncated
 - mehr als 500 Episoden erfolgreich



3. Aufbau



```
1 import gymnasium as gym
2
3 env = gym.make('CartPole-v1', render_mode="human")
4 observation, info = env.reset()
5
6 for i in range(1000):
7     action = 0 #policy: Schlitten fährt immer nach links
8     observation, reward, terminated, truncated, info = env.step(action)
9 env.close()
```

Aufgaben:

1. Mache Dich mit den Datenstrukturen in Zeile 8 vertraut.
2. Gib den Inhalt von `observation` aus.



3. action in environment



```
observation, reward, terminated, truncated, info = env.step(action)
```

Aufgaben:

3. Programmierere mit `if...`: `action = ...` eine einfache *policy* (=Strategie) um das Pendel senkrecht zu halten. Benutze dazu `observation` und `action`.



CC BY SA 4.0
Ausgenommen sind einzeln ge-kennzeichnete Inhalte/Elemente,
siehe Quellen- und Lizenzhinweise
am Ende des Dokuments.



3. RL mit TensorFlow und keras-rl2



```
1 import gym
2 import tensorflow as tf
3 from rl.agents.cem import CEMAgent
4 from rl.memory import EpisodeParameterMemory
5
6 # environment erstellen
7 env = gym.make('CartPole-v1')
8
9 # model erstellen
10 model = tf.keras.models.Sequential()
11 model.add(tf.keras.layers.Flatten(input_shape=(1,) + env.observation_space.shape))
12 model.add(tf.keras.layers.Dense(env.action_space.n))
13 model.add(tf.keras.layers.Activation('softmax'))
14
15 # agent erstellen
16 memory = EpisodeParameterMemory(limit=1000, window_length=1)
17 agent = CEMAgent(model=model, nb_actions=env.action_space.n, memory=memory,
18                 batch_size=50, nb_steps_warmup=2000, train_interval=50, elite_frac=0.05)
19 agent.compile()
20
```



```
21 # trainieren; unterbrechen mit Ctrl + C.
22 agent.fit(env, nb_steps=100000, visualize=True)
23
24 # testen
25 agent.test(env, nb_episodes=5, visualize=True)
```



Aufgaben:

4. Mache Dich mit dem Code vertraut.
5. Verändere das NN, Trainingszyklen usw.
6. Wer erreicht das beste Ergebnis?



4. Agenten können...



Aufgabe:

7. Das Lernen der Agenten ähnelt dem menschlichen Lernen - trainierte Agenten kann man dann in zahlreichen Gebieten einsetzen.

Diskutiere welche gesellschaftliche Umbrüche dies auslösen könnte.





B2 Arbeitsblätter



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.

iMINT Akademie - Fachset Informatik
KI programmieren im Unterricht Teil 4:
Reinforcement Learning
Alexander Schindler



Senatsverwaltung
für Bildung, Jugend
und Familie

BERLIN



Reinforcement Learning RL – Cart Pole

Das Pendel fällt immer wieder aus dem Gleichgewicht. Ziel: der Schlitten soll durch Ausgleichsbewegungen das Pendel möglichst senkrecht halten.

- Actionspace: Größe: _____
- Observationspace Größe: _____



[B4]

Folgende Bedingungen brechen die Ausführung ab:

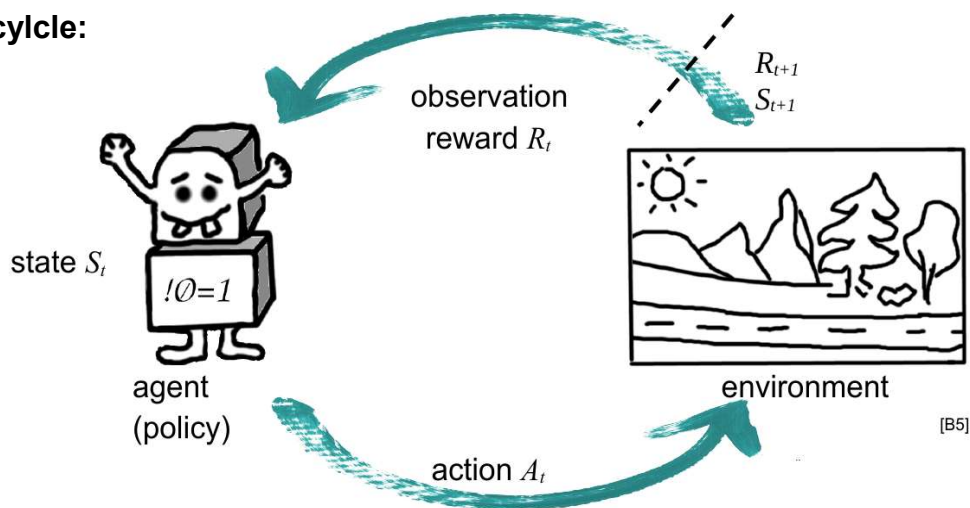
1. _____
2. _____
3. _____

[Code 1] Probelemlösung mit klassischer Programmierung:

```

1 import gymnasium as gym
2
3 env = gym.make('CartPole-v1', render_mode="human")
4 observation, info = env.reset()
5
6 for i in range(1000):
7     action = 0 # policy: Schlitten fährt immer nach links
8     observation, reward, terminated, truncated, info = env.step(action)
9     env.close()
    
```

learning cycle:



[B5]

Abb. [B4] „Cart Pole – Screenshots 2“, Alexander Schindler, Lizenz [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/), Lernaufgabe "KI im Informatikunterricht 4"

Abb. [B5] „Übersicht reinforcement learning“, Alexander Schindler, Lizenz [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/), Lernaufgabe "KI im Informatikunterricht 4"

[Code 2] Problemlösung mit RL:

```

1  import gym
2  import tensorflow as tf
3  from rl.agents.cem import CEMAgent
4  from rl.memory import EpisodeParameterMemory
5
6  # environment erstellen
7  env = gym.make('CartPole-v1')
8
9  # model erstellen
10 model = tf.keras.models.Sequential()
11 model.add(tf.keras.layers.Flatten(input_shape=(1,) +
12          env.observation_space.shape))
13 model.add(tf.keras.layers.Dense(env.action_space.n))
14 model.add(tf.keras.layers.Activation('softmax'))
15
16 # agent erstellen
17 memory = EpisodeParameterMemory(limit=1000, window_length=1)
18 agent = CEMAgent(model=model, nb_actions=env.action_space.n,
19          memory=memory, batch_size=50, nb_steps_warmup=2000,
20          train_interval=50, elite_frac=0.05)
21 agent.compile()
22
23 # trainieren; unterbrechen mit Ctrl + C.
24 agent.fit(env, nb_steps=100000, visualize=True)
25
26 # testen
27 agent.test(env, nb_episodes=5, visualize=True)

```

Aufgaben:

1. Mache Dich mit den Datenstrukturen in [Code 1] Zeile 8 vertraut.
2. Gib den Inhalt von [Code 1] `observation` aus.
3. Programmiere in [Code 1] mit `if...: action = ...` eine einfache *policy* (=Strategie) um das Pendel senkrecht zu halten. Benutze dazu `observation` und `action`.
4. Mache Dich mit [Code 2] vertraut.
5. Verändere das NN, Trainingszyklen usw.
6. Wer erreicht das beste Ergebnis?

C Hinweise für die Lehrkraft

C1 Musterlösung

Arbeitsblatt

- Actionspace: Größe: 2
- Observationspace Größe: 4

Folgende Bedingungen brechen die Ausführung ab:

1. Schlittenposition außerhalb der Grenzwerte
2. Pendelwinkel zu groß
3. Pendel mehr als 500 Episoden senkrecht

Aufgabe 1

Z. B. mit einem Debugger oder einfachen print Ausgaben in der Console.

```
print (terminated)
print (type(terminated))
```

Aufgabe 2

```
print (observation)
```

Aufgabe 3

Die einfachste policy wären zufällige Aktionen

```
# Zufällige Aktionen:
action = env.action_space.sample()
```

Mit einer einfachen If-Abfrage verbessern sich die Aktionen. Das Pendel kann aber weiterhin nicht dauerhaft senkrecht gehalten werden.

```
# Steuerung
if observation[0] > 0.00001:
    action = 0
if observation[0] < -0.00001:
    action = 1
```

Aufgabe 4

Evtl. gemeinsame Besprechung der einzelnen Bestandteile.

Aufgabe 5

ein vergrößertes NN könnte so aussehen: mit folgendem NN habe ich sehr gute Ergebnisse (Pendel wurde 500 Schritte senkrecht gehalten) erzielt:

```
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Flatten(input_shape=(1, ) +
env.observation_space.shape))
model.add(tf.keras.layers.Dense(4))
model.add(tf.keras.layers.Dense(16))
model.add(tf.keras.layers.Dense(4))
model.add(tf.keras.layers.Dense(env.action_space.n))
model.add(tf.keras.layers.Activation('softmax'))
```

Aufgabe 6

Eigenständiges experimentieren der Schüler mit vergrößertem NN und anderen Parametern ab Zeile [17]. Hinweis: Das Ausschalten der Visualisierung (`visualize=False`) beschleunigt das Training.

C2 Hinweise zur Umsetzung im Unterricht

- **Installationhinweise:** Die Bibliotheken sollten mit *pip* über die Kommandozeile installiert werden. Wichtig hier: Häufig sind mehrere Pythoninstanzen auf einem Betriebssystem installiert. Deshalb muss darauf geachtet werden dass das richtige *pip* verwendet wird. Die Erstellung eines *environments* kann hier hilfreich sein. Im Folgenden die *pip* Befehle:

```
pip install tensorflow==2.5.0
pip install gym==0.17.2
pip install keras-rl2== 1.0.5
pip install gymnasium==0.26.1
pip install gymnasium[classic_control]
```

- Sollte folgender Laufzeitfehler auftreten:
ImportError: cannot import name '__version__' from 'tensorflow.keras'...
kann er durch das Einfügen folgender Zeilen zu Beginn des Codes behoben werden:

```
tf.keras.__version__ = tf.__version__
```



C3 Didaktischer Kommentar

Ziel der Lernaufgabe ist es, den Schülern einen Einstieg in die Programmierung von *reinforcement learning* RL zu ermöglichen.

Der Code ist so einfach wie möglich gehalten, damit der Einstieg in die Programmierung mit TensorFlow leichter fällt und auch im Informatikunterricht der Sek I möglich ist. Einige Vorerfahrungen mit NN, python, Softwareentwicklung und einer IDE sollten aber vorhanden sein, somit ist die Aufgabe für Schüler der Sek II besser geeignet. Der Code bietet Anknüpfungspunkte zum Arbeiten mit Schleifen, if-Abfragen und evtl. Listen. Er lässt sich somit auch für den Anfängerunterricht einsetzen. Die Aufgabe ist skalierbar, d.h. sie kann um eigene Ideen (z. B. Problemstellungen bzw. *environments* wie sie auf www.gymlibrary.dev zu finden sind) erweitert werden und sie kann und soll somit Basis für eigene Projekte sein.

Nicht zuletzt ist mit der Thematisierung von KI ein Ausblick auf die Auswirkungen von KI auf die eigene Lebenswelt der Schüler möglich und somit ist die Aufgabe auch im Bereich Informatik und Gesellschaft zu verorten.

Lernervoraussetzungen	<ul style="list-style-type: none"> • Die SuS können einfache Algorithmen in Python programmieren. • oder: besitzen vertiefte Programmierkenntnisse in einer anderen Programmiersprache • Die SuS können mit Variablen umgehen. • Die SuS haben eine grundlegende Vorstellung von Algorithmen und Datenstrukturen. • Die SuS können eine IDE benutzen. • Die SuS haben ein grundsätzliches Verständnis für ML wie in der Lernaufgabe: „KI programmieren im Informatikunterricht Teil I: Einführung“
-----------------------	--

C4 Bezüge zum Rahmenlehrplan Informatik

Kompetenzen	Standards (Die Schülerinnen und Schüler können....)
Mit Fachwissen umgehen	Bezug zum RLP Sek I: Kompetenzbereich: Informatisches Modellieren Kompetenz: Informatische Modelle analysieren und bilden



CC BY SA 4.0
 Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.





	<p>Standard F: informatische Modelle als reduzierte Abbildung der realen Welt beschreiben und beurteilen</p> <p>Standard G: ein Modell selbst erstellen</p>
Erkenntnisse gewinnen	Die SuS können Modelle selbst erstellen und mit NN umgehen.
Kommunizieren	<p>Bezug zum RLP Sek I:</p> <p>Kompetenzbereich: Kommunizieren und Kooperieren – Teamarbeit organisieren und koordinieren</p> <p>Kompetenz: Arbeitsergebnisse dokumentieren und präsentieren</p> <p>Standard G: adressatengerecht mit Softwareunterstützung präsentieren</p>
Bewerten	<p>Bezug zum RLP Sek I:</p> <p>Kompetenzbereich: Informatisches Modellieren</p> <p>Kompetenz: Informatische Modelle analysieren und bilden</p> <p>Standard F: informatische Modelle als reduzierte Abbildung der realen Welt beschreiben und beurteilen</p> <p>Standard H: beurteilen, ob das selbst erstellte Modell problemadäquat ist</p>

Unterrichtsfach	Informatik
Jahrgangsstufe/n	Sek I: 10 Sek II: IN-3
Niveaustufe/n	<p>Bezug zum RLP Sek I:</p> <p>Kompetenzbereich: Informatisches Modellieren</p> <p>Kompetenz: Informatische Modelle analysieren und bilden</p> <p>Standard F: informatische Modelle als reduzierte Abbildung der realen Welt beschreiben und beurteilen</p> <p>Standard G: ein Modell selbst erstellen</p> <p>Standard H: beurteilen, ob das selbst erstellte Modell problemadäquat ist</p>



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.





	Bezug zum RLP Sek II: 3. Kurshalbjahr (IN-3) Grundlagen der Informatik und Vertiefungsgebiet: V5 Künstliche Intelligenz
Zeitraumen	1. Doppelstunde: Einstieg / Theorie / Aufgaben 1-3. 2. Doppelstunde: Aufgaben 4-7. 3. Doppelstunde und weitere: Eigenes Projekt
Thema	

Kontext	<ul style="list-style-type: none"> • Beruf und Arbeitswelt • Robotik • autonome Agenten • Gesellschaft
Schlagwörter	Programmierung, Informatik und Gesellschaft, Informatiksystem verstehen, Problemlösen, mit Informationen umgehen, Modellbildung, künstliche Intelligenz, KI, Deep Learning, DL, TensorFlow, Keras, Python, Daten

C5 Bezüge zum Basiscurriculum Medienbildung¹

Standards des BC Medienbildung	Die Schülerinnen und Schüler können ...
Präsentieren	<ul style="list-style-type: none"> • Arbeitsergebnisse vorstellen • Reflektierter Technologieeinsatz
Reflektieren	<ul style="list-style-type: none"> • Chancen und Risiken von Geschäftsaktivitäten im Internet untersuchen und Schlussfolgerungen für eigene Geschäftsaktivitäten ziehen. • Die Möglichkeiten und Methoden medialer Manipulation exemplarisch analysieren.

¹ vgl. Rahmenlehrplan Jahrgangsstufen 1-10, Teil B, S. 15-22, Berlin, Potsdam 2015



CC BY SA 4.0
 Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.



Senatsverwaltung
 für Bildung, Jugend
 und Familie





C6 Bezüge zu übergreifenden Themen²

Berufs-und Studienorientierung	Data Engineer, Data Analyst, Data Scientist, Machine Learning Engineer, Softwareentwickler
Verbraucherbildung	Präsenz von neuronalen Netzen und RL in alltäglichen Geräten, Anwendungen

C6 Bezüge zu anderen Fächern:

- Gesellschaftswissenschaften: Einsatz von RL im Bereich von Finanzen (z. B. Unterstützung bei Anlageentscheidungen), Spiele (z. B. AlphaGo), Logistik (z. B. Routenoptimierung) usw.

² vgl. Rahmenlehrplan Jahrgangsstufen 1-10, Teil B, S. 24ff, Berlin, Potsdam 2015



CC BY SA 4.0
Ausgenommen sind einzeln gekennzeichnete Inhalte/Elemente, siehe Quellen- und Lizenzhinweise am Ende des Dokuments.



D Anhang

D3 Quellen / Lektüreliste zum Weiterlesen

- MORONEY, Laurence[2020]: AI and Machine Learning for Coders, A Programmer's Guide to Artificial Intelligence.
- WINDER, Phil [2020], Reinforcement Learning: Industrial Applications of Intelligent Agents
- Vergleich verschiedener reinforcement learning frameworks von Phil Winder abgerufen am (1.12.2023):
<https://winder.ai/a-comparison-of-reinforcement-learning-frameworks-dopamine-rl-lib-keras-rl-coach-trfl-tensorforce-coach-and-more/>
- Gym API: www.gymnasium.dev
- keras-rl2: <https://github.com/inarikami/keras-rl2>

D4 Bildnachweise / Abbildungen

Bildtitel	Seite	Quelle
[B1]	1	„AAEOR“, Alexander Schindler, Lizenz CC BY-SA 4.0 , Lernaufgabe "KI im Unterricht 4"
[B2]	8	„We visited Zoom Torino : my daughter Chloé enjoys the zoo playground ! (Turin / Italy - 08/2015) #zoo #zoomtorino #bioparc #playground #Torino #Turin #child #kids #enfant #travelwithkids #familytravel #familytravel #familytrip #cute #happy #smile #fun We", by leszed.com, Lizenz CC BY-SA 2.0 , flickr , abgerufen am: 10.7.2024
[B3]	11	„Cart Pole – Screenshots 1“, Alexander Schindler, Lizenz CC BY-SA 4.0 , Lernaufgabe "KI im Informatikunterricht 4"
[B4]	12, 13, 21	„Cart Pole – Screenshots 2“, Alexander Schindler, Lizenz CC BY-SA 4.0 , Lernaufgabe "KI im Informatikunterricht 4"
[B5]	9, 21	„Übersicht reinforcement learning“, Alexander Schindler, Lizenz CC BY-SA 4.0 , Lernaufgabe "KI im Informatikunterricht 4"