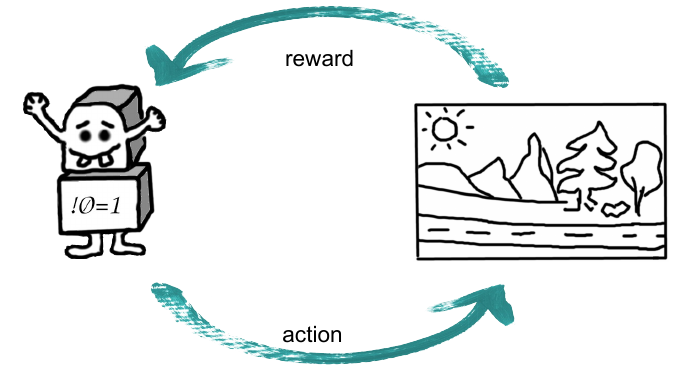


# KI programmieren im Informatikunterricht Teil 4

**Reinforcement Learning**



[B1]

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

|  |  |
| --- | --- |
|  | [C:\Users\kube\AppData\Local\Microsoft\Windows\INetCache\Content.Word\sdg_4.jpgsdg_8sdg_9](https://17ziele.de/) |

Inhaltsverzeichnis

[KI programmieren im Informatikunterricht Teil 4 1](#__RefHeading___Toc7484_2681379757)

[A Überblick 3](#__RefHeading___Toc7316_2681379757)

[A 1 Zusammenfassung: 4](#__RefHeading___Toc6155_2319361435)

[A2 Stundenübersicht 5](#__RefHeading___Toc7318_2681379757)

[A3 Themeneinstieg und theoretische Grundlagen 6](#__RefHeading___Toc4008_1635008047)

[B Lernaufgabe 7](#__RefHeading___Toc4010_1635008047)

[B1 Unterrichtsbegleitende Präsentation 7](#__RefHeading___Toc2235_3609271763)

[B2 Arbeitsblätter 20](#__RefHeading___Toc2237_3609271763)

[C Hinweise für die Lehrkraft 23](#__RefHeading___Toc4012_1635008047)

[C1 Musterlösung 23](#__RefHeading___Toc6193_23193614351)

[C2 Hinweise zur Umsetzung im Unterricht 25](#__RefHeading___Toc6193_231936143511)

[C3 Didaktischer Kommentar 26](#__RefHeading___Toc6193_2319361435)

[C4 Bezüge zum Rahmenlehrplan Informatik 26](#__RefHeading___Toc6195_2319361435)

[C5 Bezüge zum Basiscurriculum Medienbildung 28](#__RefHeading___Toc6197_2319361435)

[C6 Bezüge zu übergreifenden Themen 28](#__RefHeading___Toc6199_2319361435)

[C6 Bezüge zu anderen Fächern: 29](#__RefHeading___Toc6201_2319361435)

[D Anhang 30](#__RefHeading___Toc4014_1635008047)

[D3 Quellen / Lektüreliste zum Weiterlesen 30](#__RefHeading___Toc7454_2681379757)

[D4 Bildnachweise / Abbildungen 30](#__RefHeading___Toc7456_2681379757)

# A Überblick

|  |  |
| --- | --- |
| Unterrichtsfach | Informatik |
| Jahrgangsstufe(n) | 10-13 |
| Niveaustufe | verschiedene (siehe C4 Bezüge zum Rahmenlehrplan Informatik) |
| Zeitrahmen | Zwei Doppelstunden |
| Thema | Künstliche Intelligenz |
| Themenfeld(er) |  |
| Kontext |  |
| Zusammenfassung  des Inhalts | Wer neuronale Netze einmal selbst trainiert hat, kann deren Risiken, Probleme und Chancen und damit auch mögliche gesellschaftliche Entwicklungen besser einschätzen.  In dieser Lernaufgabe lernen die Schüler:innen den Ablauf beim Reinforcement Learning RL. Im Anschluss ist es den Schüler:innen möglich, eigene Projekte durchzuführen.  Diese Lernaufgabe setzt die Grundlagen der ersten Lernaufgabe: „KI programmieren im Informatikunterricht Teil I: Einführung“ voraus. In diesem Sinne: *All hands on code!* |
| Didaktischer Kommentar | Ziel der Lernaufgabe ist es, SuS einen Einstieg in die Programmierung von reinforcement learning RL zu ermöglichen.  Der Code ist so einfach wie möglich gehalten. Die Aufgabe ist skalierbar, d.h. sie kann um eigene Ideen (z. B. *environments* auf [www.gymlibrary.dev](https://www.gymlibrary.dev/)) erweitert werden und sie kann und soll somit Basis für eigene Projekte sein.  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. |
| 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 |
|  | * Schüler und Schülerinnen benötigen eine entsprechend leistungsfähige Pyhton 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 Asführen eine eher ältere Pythonversion zu verwenden. * Alle Bibilotheken lassen sich über pip install <Bibliothek> installieren. Am besten unten stehende Reihenfolge einhalten. **Folgende Bibliotheken müssen in der angegebenen Version installiert sein:**   + 1. **tensorflow == 2.5.0**     2. **gym== 0.17.2**     3. **keras-rl2==1.0.5**     4. **gymnasium==0.26.2**     5. **gymnasium[classic\_control])** |

## 

## A 1 Zusammenfassung:

**Intention der Lernaufgabe**

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

**Hinweise:**

Das Zusammenspiel der Bibliotheken TensorfFlow, 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:

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 (*action*) auf einem Spielplatz (*environment*) und fühlt sich glücklich (*reward*). Geht es morgen wieder schaukeln?  Impuls zur Bedeutung und Problematik von selbst lernenden Sysmten wie z. B. Robotik, Spiele | Präsentation |
| 10 Min. | Vortrag / LSG | *Lehrervortrag zu Grundlagen des reinforcement learnings* | Präsentation |
| 10 Min. |  | Besprechung Quellcode |  |
| 55 Min. | Erarbeitung | Aufgaben 1-3 |  |
| 10 Min. |  | Diskussion der Ergebnisse |  |

*LSG: Lehrer-Schüler-Gespräch*

**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).

# B Lernaufgabe

## B1 Unterrichtsbegleitende Präsentation

**1. Reinforcement Learning RL**

****

Schaukeln macht Spaß.

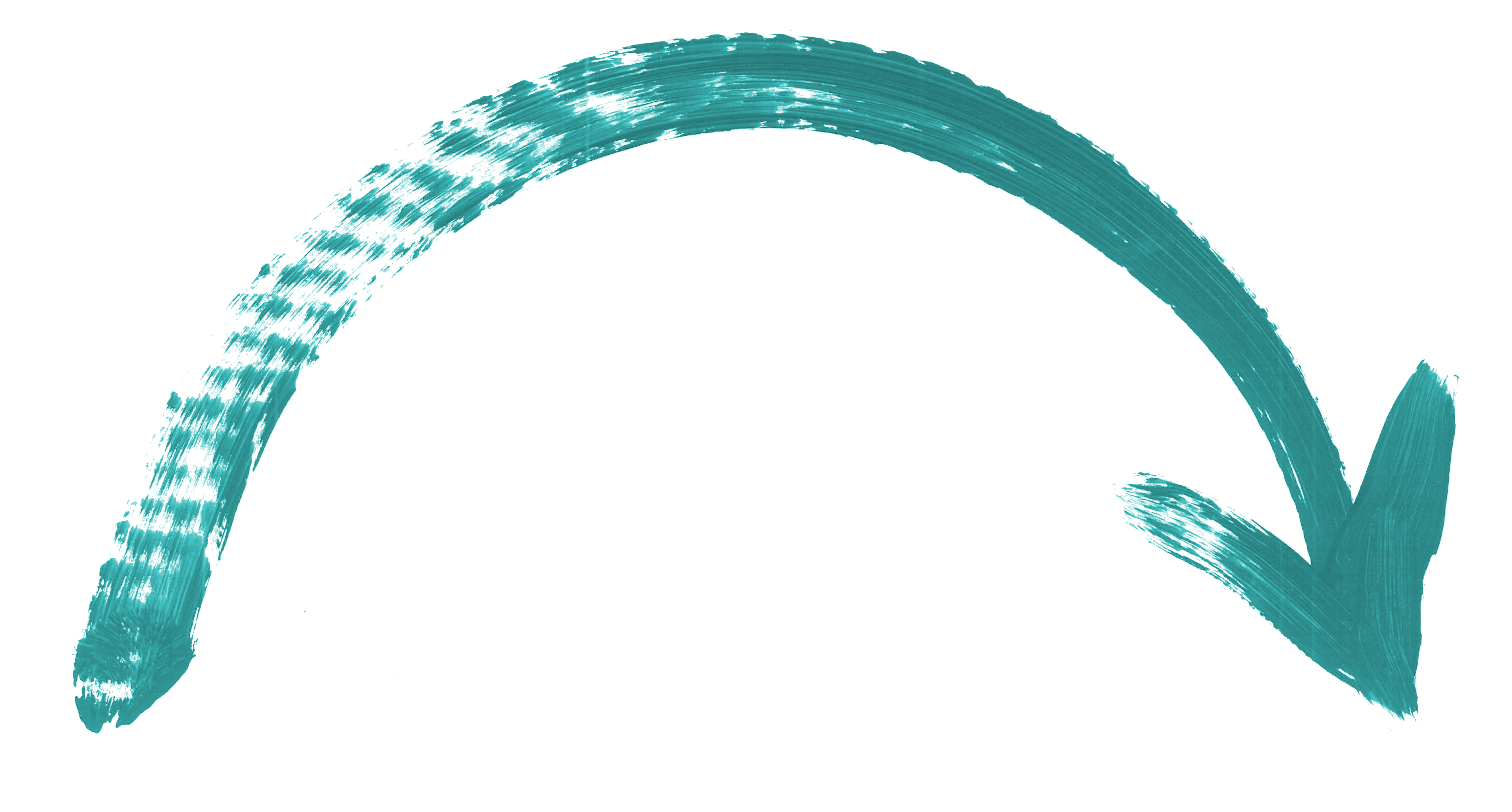
Gehen wir morgen wieder

auf den Spielplatz?

[B2]

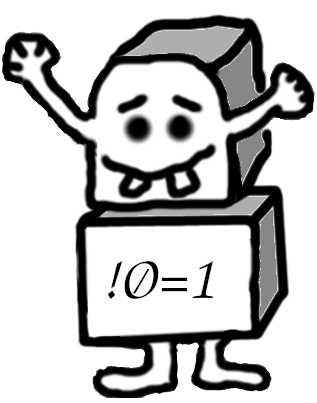
Abb. [B2]:„We visited Zoom Torino : my daughter Chloé enjoys the zoo playground ! (Turin / Italy - 08/2015) #zoo #zoomtorino #bioparc #playground #Torino #Turin #child #kids #enfant #travelwithkids #familytravel #familytravel #familytrip #cute #happy #smile #fun We", by leszed.com, [Lizenz CC BY-SA 2.0](https://creativecommons.org/licenses/by-sa/2.0/?ref=openverse), [flickr](https://www.flickr.com/photos/131857276@N03/21090199171)

**1. RL - learning cycle**

****

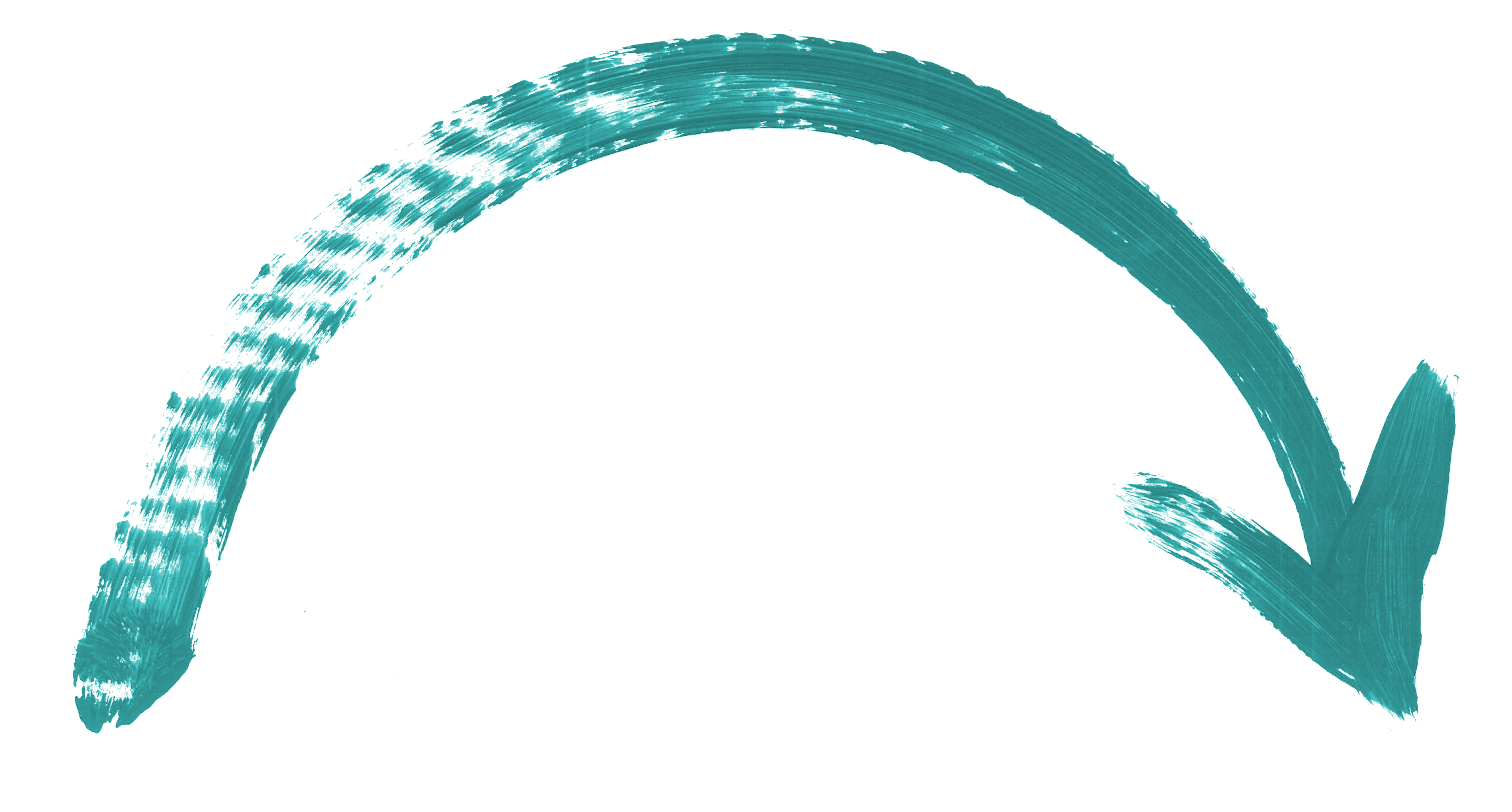
*Rt+1*

*St+1*

observation

reward *Rt*

state *St*

 agent environment

(policy)

[B5]

action *At*

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

**1. AREA**

**A** – Action

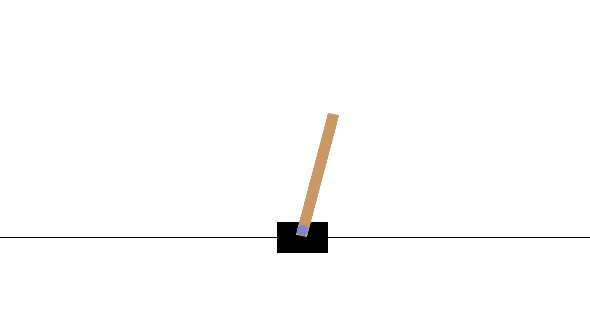
**R** – Reward

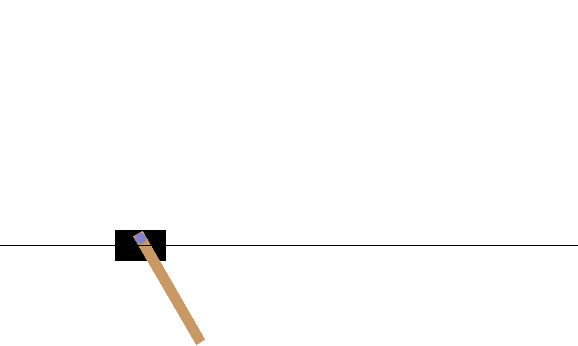
**E** – Environment

**A** – Agent

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

**1. Reinforcement Learning**

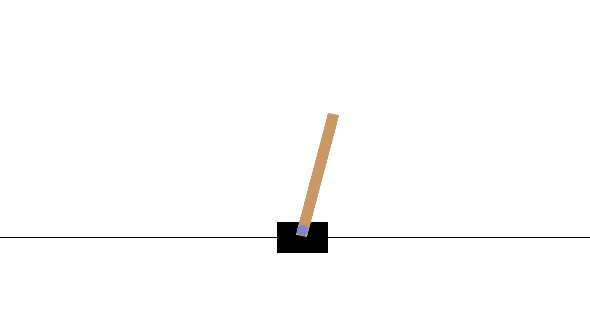
****

* klassisches Problem: CartPol
* Umgebung: gymnasium / gym
* Framework: keras-rl2
* Neuronales Netzwerk: TensorfFlow

[B3]

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

**2. CartPole**



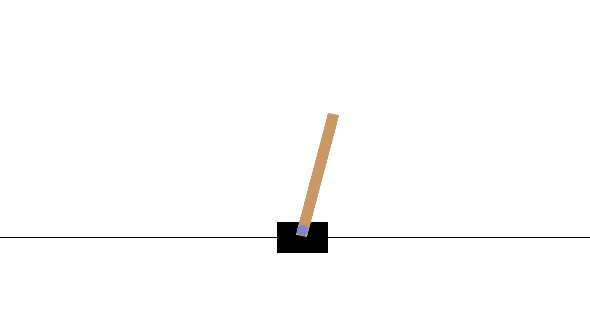
* Pendel senkrecht balancieren
* *action space*: Schlitten bewegen

[B4]

* *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/deed.de), Lernaufgabe "KI im Informatikunterricht 4“

**2. CartPole - *spaces***



* *action space*
* Schlitten nach links action = 0
* Schlitten nach rechts action = 1

[B4]

* *observation space*

[0] Schlittenposition -4.8 - 4.8

[1] Schlittengeschwindigkeit -Inf - Inf

[2] Pendelwinkel -0.418rad (-24°) - 0.418rad (24°)

[3] Pendelwinkelgeschwindigkeit -Inf – Inf

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

**2. CartPole - Abbruchkriterien**

* terminated
  + - * Schlittenposition: observation[0] > ±2.4
      * Pendelwinkel: observation[2] > ±12°
* truncated
  + - * mehr als 500 Episoden erfolgreich

**3. Aufbau**

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

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

**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. *# environment erstellen*
6. env = gym.make('CartPole-v1')
8. *# model erstellen*
9. **model** = tf.keras.models.Sequential()
10. **model**.add(tf.keras.layers.Flatten(input\_shape=(1,) + env.observation\_space.shape))
11. **model**.add(tf.keras.layers.Dense(env.action\_space.n))
12. **model**.add(tf.keras.layers.Activation('softmax'))
13. *# agent erstellen*
14. memory = EpisodeParameterMemory(limit=1000, window\_length=1)
15. **agent** = CEMAgent(model=**model**, nb\_actions=env.action\_space.n, memory=memory,
16. batch\_size=50, nb\_steps\_warmup=2000, train\_interval=50, elite\_frac=0.05)
17. **agent**.compile()
18. *# trainieren; unterbrechen mit Ctrl + C.*
19. **agent**.fit(env, nb\_steps=100000, visualize=True)
20. *# testen*
21. **agent**.test(env, nb\_episodes=5, visualize=True)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Aufgaben:**

1. Mache Dich mit dem Code vertraut.
2. Verändere das NN, Trainingszyklen usw.
3. Wer erreicht das beste Ergebnis?

**4. Agenten können...**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

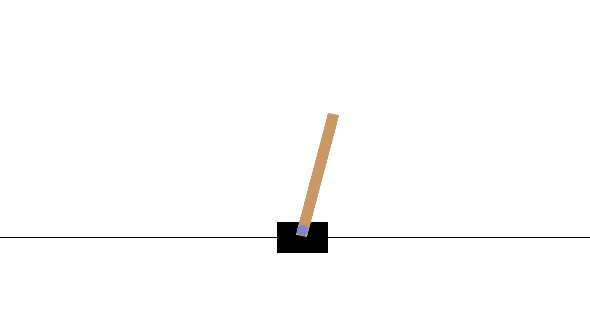
**Aufgabe:**

1. 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

**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: \_\_\_\_\_\_\_\_\_\_\_\_

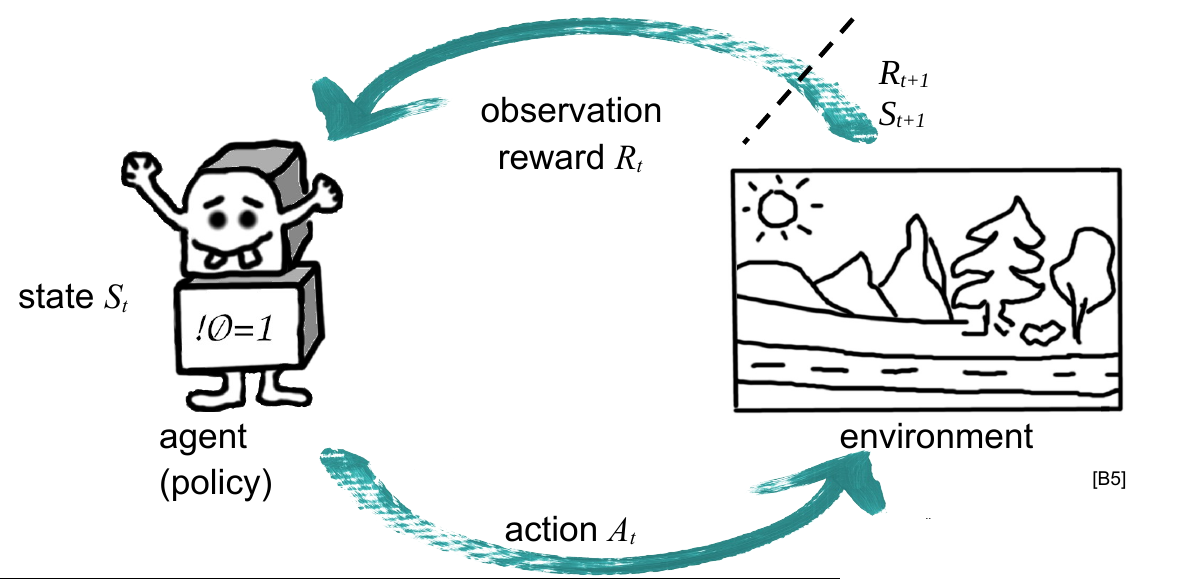
Folgende Bedingungen brechen die Ausführung ab:

[B4]

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**[Code 1] Probelemlösung mit klassischer Programmierung:**

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



**learning cylcle:**

Abb. [B4] „Cart Pole – Screenshots 2“, Alexander Schindler, Lizenz [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/deed.de), 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/deed.de), 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. *# environment erstellen*
6. env = gym.make('CartPole-v1')
8. *# model erstellen*
9. **model** = tf.keras.models.Sequential()
10. **model**.add(tf.keras.layers.Flatten(input\_shape=(1,) +
11. env.observation\_space.shape))
12. **model**.add(tf.keras.layers.Dense(env.action\_space.n))
13. **model**.add(tf.keras.layers.Activation('softmax'))
14. *# agent erstellen*
15. memory = EpisodeParameterMemory(limit=1000, window\_length=1)
16. **agent** = CEMAgent(model=model, nb\_actions=env.action\_space.n,
17. memory=memory, batch\_size=50, nb\_steps\_warmup=2000,
18. train\_interval=50, elite\_frac=0.05)
19. **agent**.compile()
20. *# trainieren; unterbrechen mit Ctrl + C.*
21. **agent**.fit(env, nb\_steps=100000, visualize=True)
22. *# testen*
23. **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 experimenterien 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 Betriebssytem 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. *einvironments* wie sie auf [www.gymlibrary.dev](https://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.

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

|  |  |
| --- | --- |
| Unterrichtsfach | Informatik |
| Jahrgangsstufe/n | Sek I: 10  Sek II: IN-3 |
| Niveaustufe/n | **Bezug zum RLP Sek I:**  **Kompetenzbereich**: Informatisches Modellieren  **Kompetenz**: Informatische Modelle analysieren und bilden  **Standard F**: informatische Modelle als reduzierte Abbildung der realen Welt beschreiben und beurteilen  **Standard G**: ein Modell selbst erstellen  **Standard H**: beurteilen, ob das selbst erstelle Modell problemadäquat ist  **Bezug zum RLP Sek II:**  3. Kurshalbjahr (**IN-3**) Grundlagen der Informatik und Vertiefungsgebiet: V5 Künstliche Intelligenz |
| Zeitrahmen | **1. Doppelstunde:** Einstieg / Theorie / Aufgaben 1-3.  **2. Doppelstunde**: Aufgaben 4-7.  **3. Doppelstunde und weitere:** Eigenes Projekt |
| Thema |  |

|  |  |
| --- | --- |
| Kontext | * 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 Medienbildung[[1]](#footnote-1)

|  |  |
| --- | --- |
| **Standards des BC Medienbildung** | Die Schülerinnen und Schüler können … |
| Präsentieren | * Arbeitsergebnisse vorstellen * Reflektierter Technologieeinsatz |
| Reflektieren | * Chancen und Risiken von Geschäftsaktivitäten im Internet untersuchen und Schlussfolgerungen für eigene Geschäftsaktivitäten ziehen. * Die Möglichkeiten und Methoden medialer Manipulation exemplarisch analysieren. |

## C6 Bezüge zu übergreifenden Themen[[2]](#footnote-2)

|  |  |
| --- | --- |
| 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. Routenoptiermierung) usw.

# 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-rllib-keras-rl-coach-trfl-tensorforce-coach-and-more/>

* Gym API: [www.gymlibrary.dev](http://www.gymlibrary.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](https://creativecommons.org/licenses/by-sa/4.0/deed.de), 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](https://creativecommons.org/licenses/by-sa/2.0/?ref=openverse), [flickr](https://www.flickr.com/photos/131857276@N03/21090199171), abgerufen am: 10.7.2024 |
| [B3] | 11 | „Cart Pole – Screenshots 1“, Alexander Schindler, Lizenz [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/deed.de), Lernaufgabe "KI im Informatikunterricht 4“ |
| [B4] | 12, 13, 21 | „Cart Pole – Screenshots 2“, Alexander Schindler, Lizenz [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/deed.de), Lernaufgabe "KI im Informatikunterricht 4“ |
| [B5] | 9, 21 | „Übersicht reinforcement learning“, Alexander Schindler, Lizenz [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/deed.de), Lernaufgabe "KI im Informatikunterricht 4“ |

1. vgl. Rahmenlehrplan Jahrgangsstufen 1-10, Teil B, S. 15-22, Berlin, Potsdam 2015 [↑](#footnote-ref-1)
2. vgl. Rahmenlehrplan Jahrgangsstufen 1-10, Teil B, S. 24ff, Berlin, Potsdam 2015 [↑](#footnote-ref-2)