

Image credit: [Volvo cars](https://www.volvocars.com)

Künstliche Intelligenz (KI) in SCHULEN

Projekt: Calliope-Rennspiel

Wissen zu Künstlicher Intelligenz spielerisch in die Schulen bringen

- SuS-fokussierte Präsentation für Workshops -

Ferenc Hechler, Mirko Jelinek, Christian Schiller, Dirk Wolters



LIFE IS FOR SHARING.

[AI@SCHOOL](https://ai@school.de) – [Autonomes Fahren mit dem Calliope mini](https://ai@school.de) - Seite 1



Dieses Material steht unter der Creative-Commons-Lizenz Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by-sa/4.0/>

Unser Gehirn



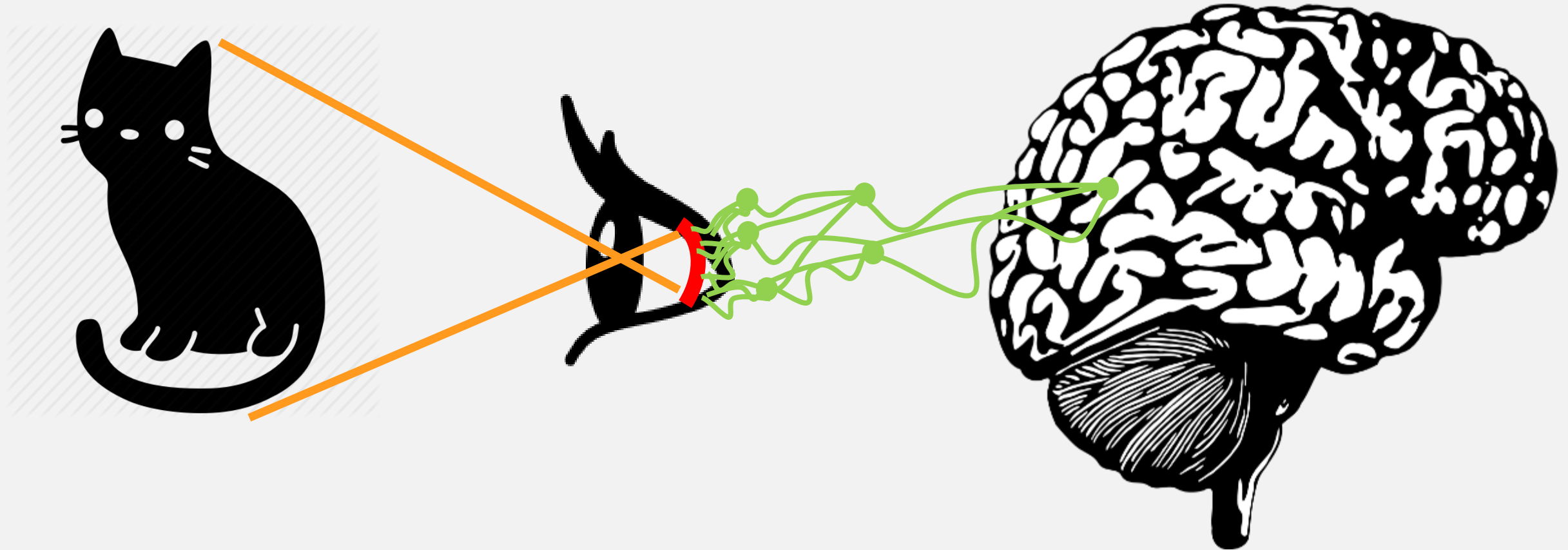
LIFE IS FOR SHARING.

[AI@SCHOOL](#) – [Autonomes Fahren mit dem Calliope mini](#) - Seite 2

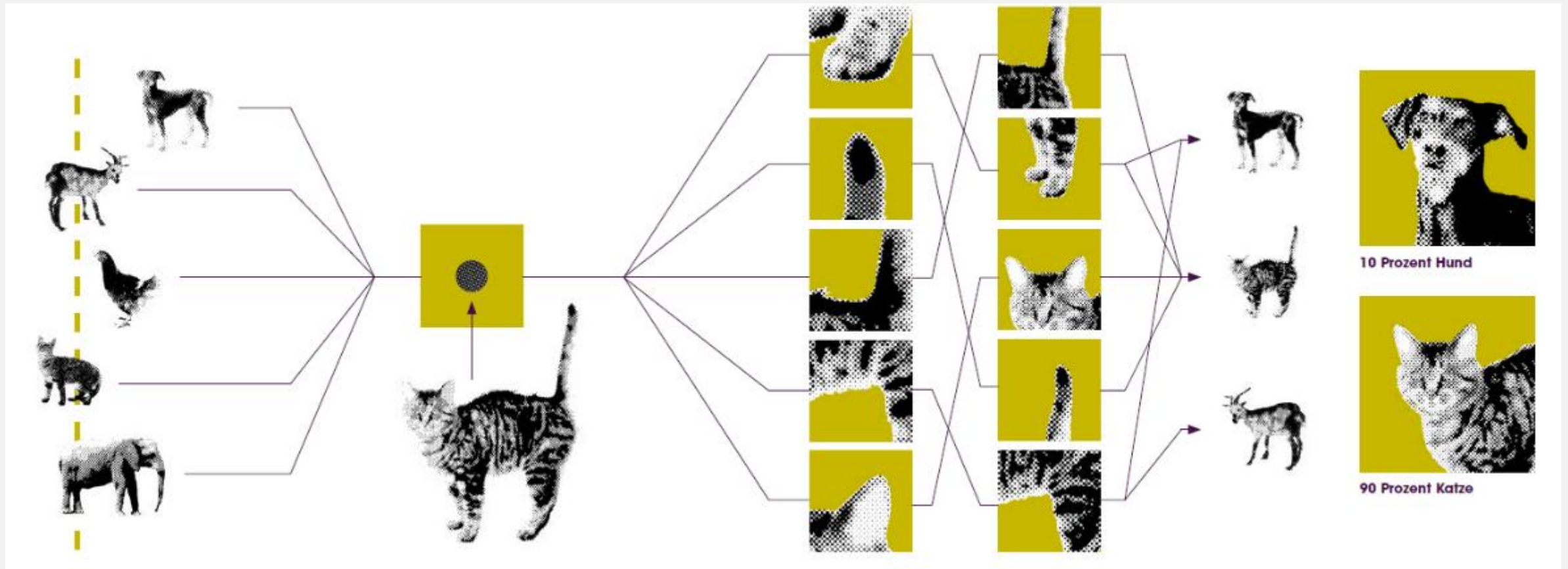


Dieses Material steht unter der
Creative-Commons-Lizenz Namensnennung - Weitergabe
unter gleichen Bedingungen 4.0 International.
Um eine Kopie dieser Lizenz zu sehen, besuchen Sie
<http://creativecommons.org/licenses/by-sa/4.0/>

Wie funktioniert unser Gehirn?



Wie neuronale Netze Katzen erkennen?



Quelle: Bundesministerium für Bildung und Forschung

Beispiele für künstliche Intelligenz



Ein klassisches Programm



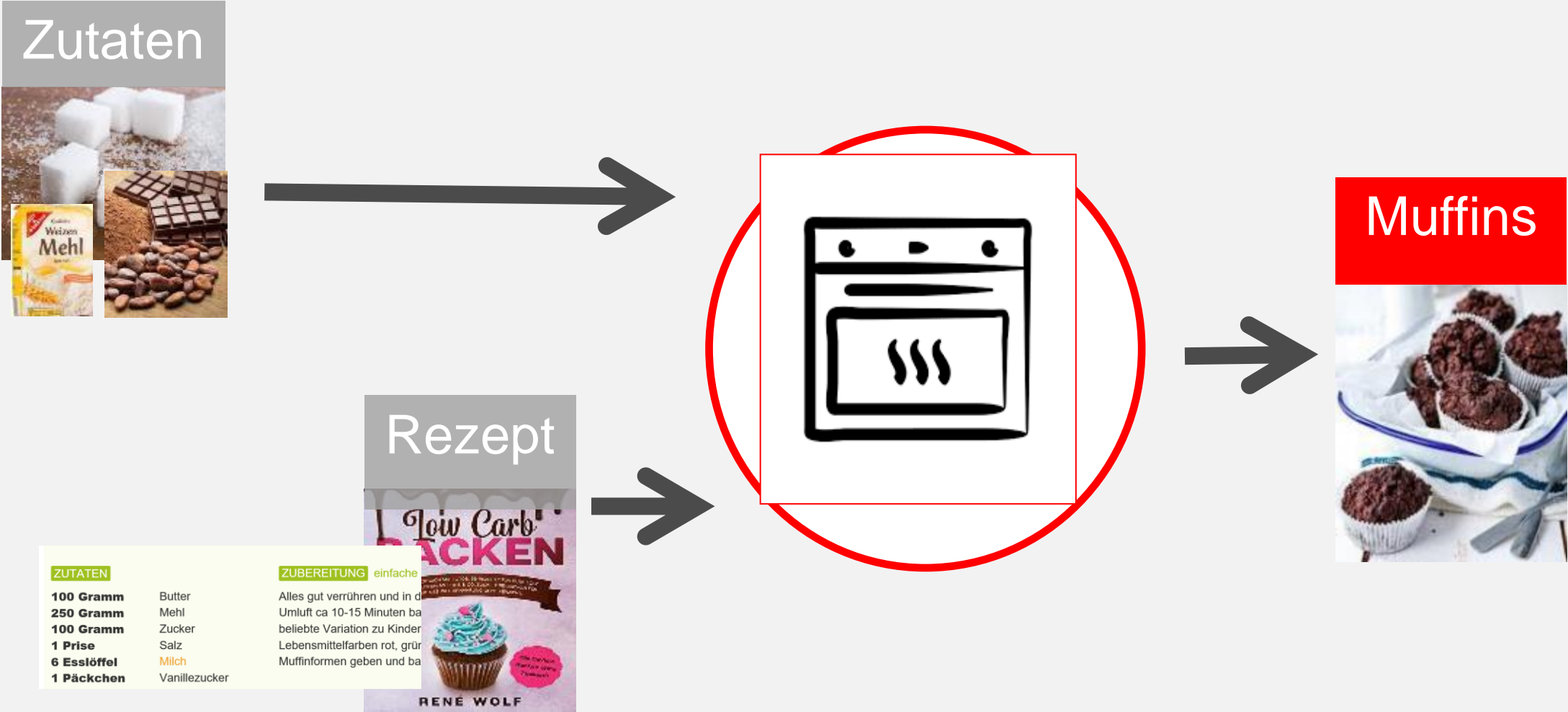
LIFE IS FOR SHARING.

[AI@SCHOOL](#) – [Autonomes Fahren mit dem Calliope mini](#) - Seite 6

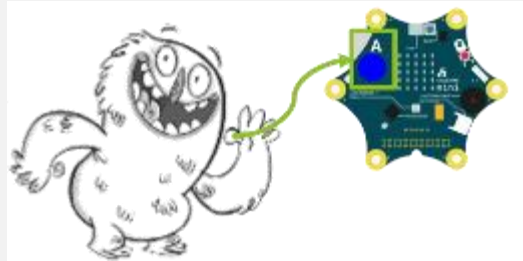


Dieses Material steht unter der
Creative-Commons-Lizenz Namensnennung - Weitergabe
unter gleichen Bedingungen 4.0 International.
Um eine Kopie dieser Lizenz zu sehen, besuchen Sie
<http://creativecommons.org/licenses/by-sa/4.0/>

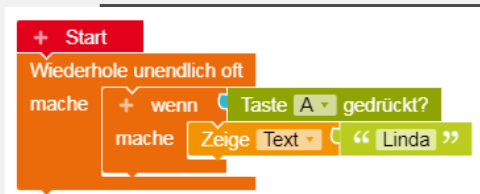
Wie würden wir backen?



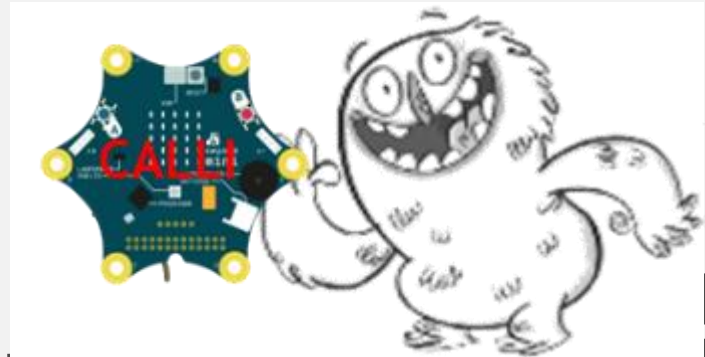
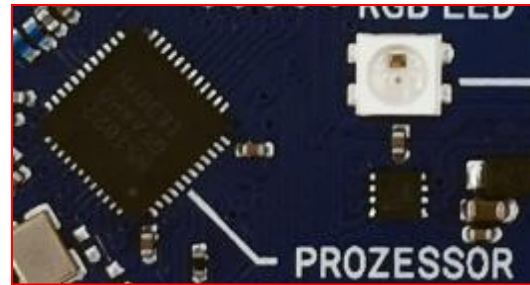
Wie funktioniert Programmierung?



Reale-Daten

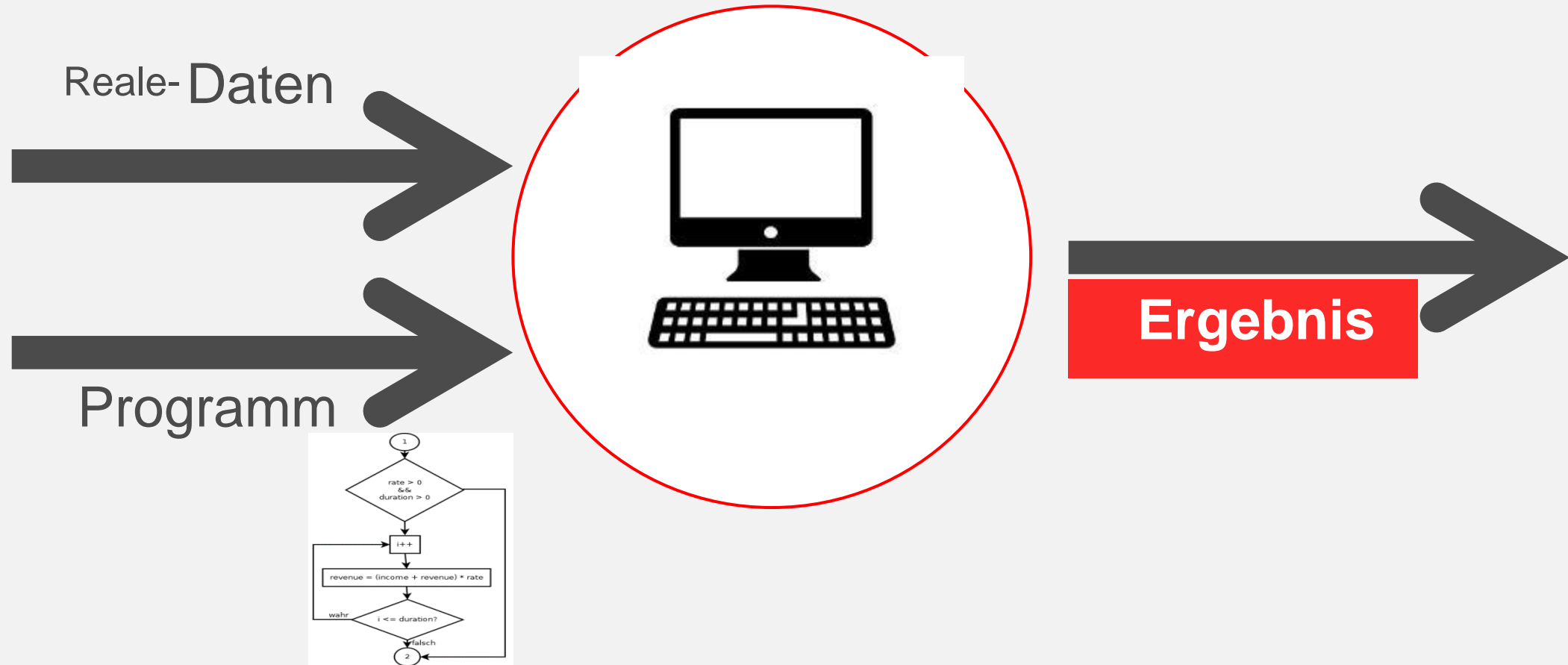


Programm

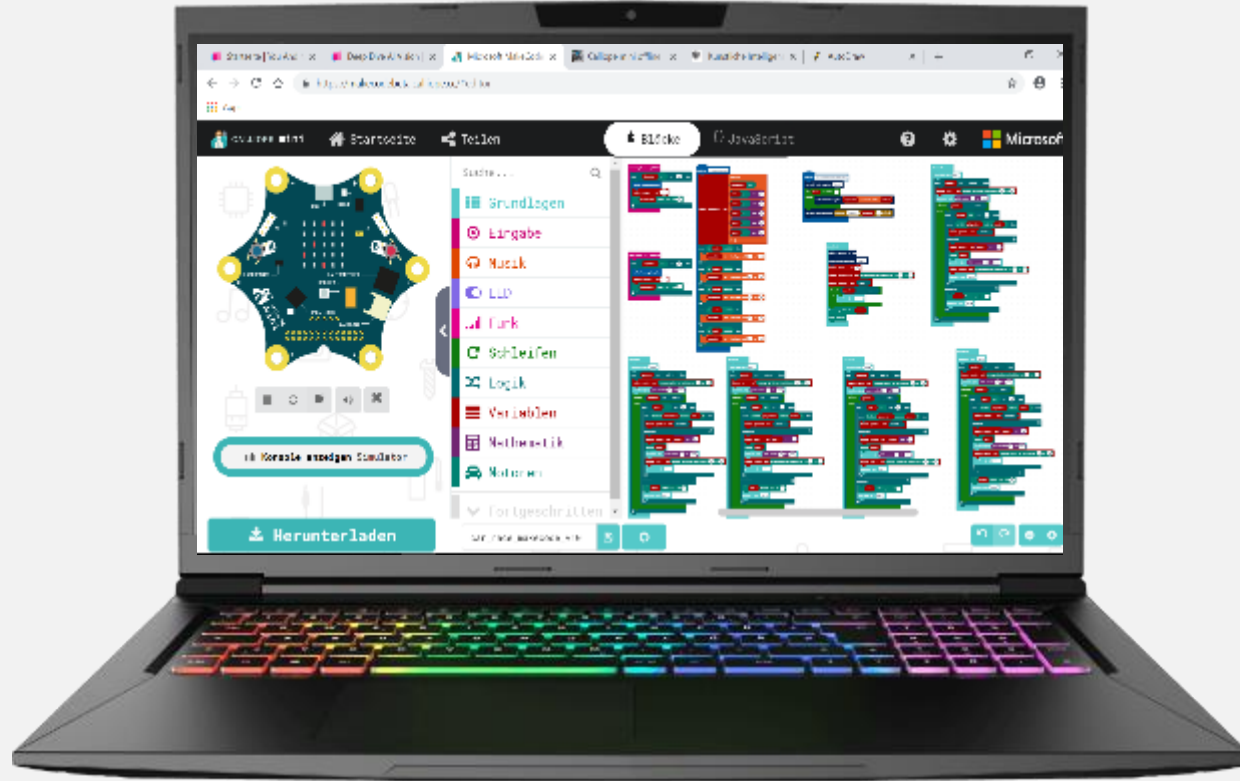


Ergebnis

Wie funktioniert Programmierung?



Was benötigen wir dafür?



Link zum Editor: <https://makecode.calliope.cc/#editor>



LIFE IS FOR SHARING.

AI@SCHOOL – [Autonomes Fahren mit dem Calliope mini](#) - Seite 10

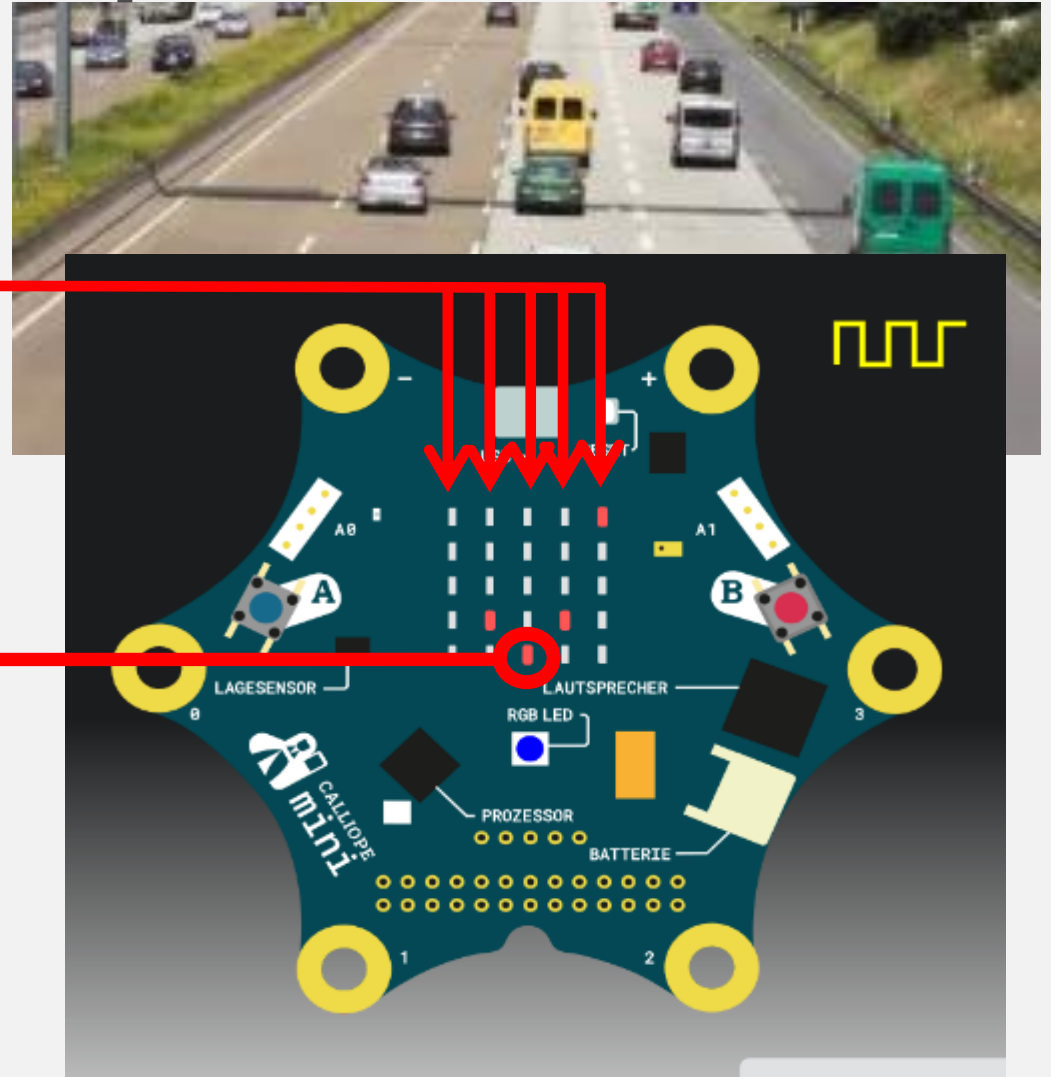


Dieses Material steht unter der Creative-Commons-Lizenz Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by-sa/4.0/>

Wie funktioniert das Rennspiel?

5 Fahrbahnen mit Autos
die überholt werden müssen

Eigenes Auto



Wie ist das Spiel aufgebaut? – Die Fahrbahnen



Scratch script for car1:

```
when green flag clicked
  position (x) 200
  wenn gesteuert == wahr dann
    setze car1 auf orange Sprites an Position x: 2 y: 2
  position (m) pick random 2 to 500
  während gesteuert == wahr
    mache
      wenn car1 y+ == 4 dann
        wenn berührt playerCar anderen car1? dann
          setze gesteuert auf falsch
        ansonsten
          setze score auf score++ 1
        lösche car1
        position (m) pick random 2 to 500
        setze car1 auf orange Sprites an Position x: 2 y: 2
        position (m) 500
      ansonsten
        car1 setze y+ um 1
        position (m) 500
```

Scratch script for car2:

```
when green flag clicked
  position (m) 300
  wenn gesteuert == wahr dann
    setze car2 auf orange Sprites an Position x: 2 y: 2
  position (m) pick random 2 to 500
  während gesteuert == wahr
    mache
      wenn car2 y+ == 2 dann
        wenn berührt playerCar anderen car2? dann
          setze gesteuert auf falsch
        ansonsten
          setze score auf score++ 1
        lösche car2
        position (m) pick random 2 to 500
        setze car2 auf orange Sprites an Position x: 2 y: 2
        position (m) 500
      ansonsten
        car2 setze y+ um 1
        position (m) 500
```

Scratch script for car3:

```
when green flag clicked
  position (m) 200
  wenn gesteuert == wahr dann
    setze car3 auf orange Sprites an Position x: 2 y: 2
  position (m) pick random 2 to 500
  während gesteuert == wahr
    mache
      wenn car3 y+ == 4 dann
        wenn berührt playerCar anderen car3? dann
          setze gesteuert auf falsch
        ansonsten
          setze score auf score++ 1
        lösche car3
        position (m) pick random 2 to 500
        setze car3 auf orange Sprites an Position x: 2 y: 2
        position (m) 500
      ansonsten
        car3 setze y+ um 1
        position (m) 500
```

Scratch script for car4:

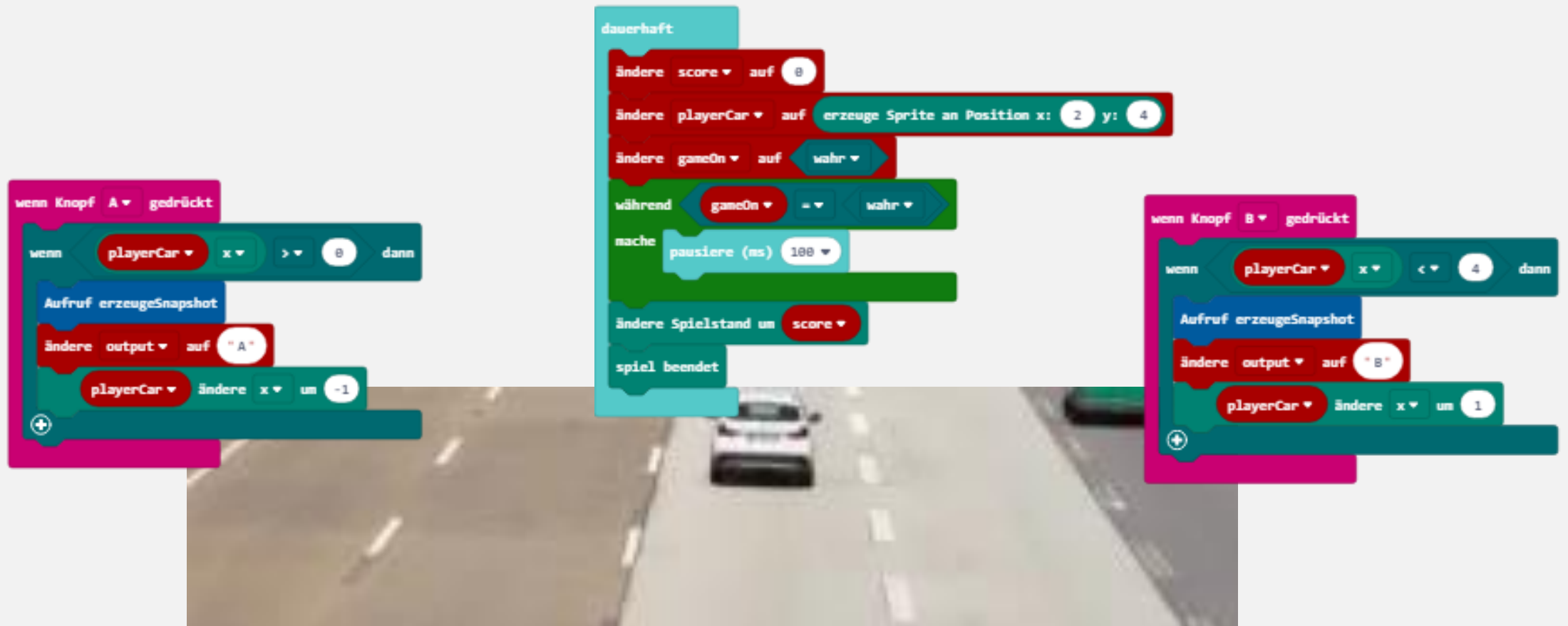
```
when green flag clicked
  position (m) 100
  wenn gesteuert == wahr dann
    setze car4 auf orange Sprites an Position x: 2 y: 2
  position (m) pick random 2 to 500
  während gesteuert == wahr
    mache
      wenn car4 y+ == 4 dann
        wenn berührt playerCar anderen car4? dann
          setze gesteuert auf falsch
        ansonsten
          setze score auf score++ 1
        lösche car4
        position (m) pick random 2 to 500
        setze car4 auf orange Sprites an Position x: 2 y: 2
        position (m) 500
      ansonsten
        car4 setze y+ um 1
        position (m) 500
```

Scratch script for car5:

```
when green flag clicked
  position (m) 100
  wenn gesteuert == wahr dann
    setze car5 auf orange Sprites an Position x: 2 y: 2
  position (m) pick random 2 to 500
  während gesteuert == wahr
    mache
      wenn car5 y+ == 2 dann
        wenn berührt playerCar anderen car5? dann
          setze gesteuert auf falsch
        ansonsten
          setze score auf score++ 1
        lösche car5
        position (m) pick random 2 to 500
        setze car5 auf orange Sprites an Position x: 2 y: 2
        position (m) 500
      ansonsten
        car5 setze y+ um 1
        position (m) 500
```



Wie ist das Spiel aufgebaut? – Eigenes Auto



Maschinelles Lernen - KI



LIFE IS FOR SHARING.

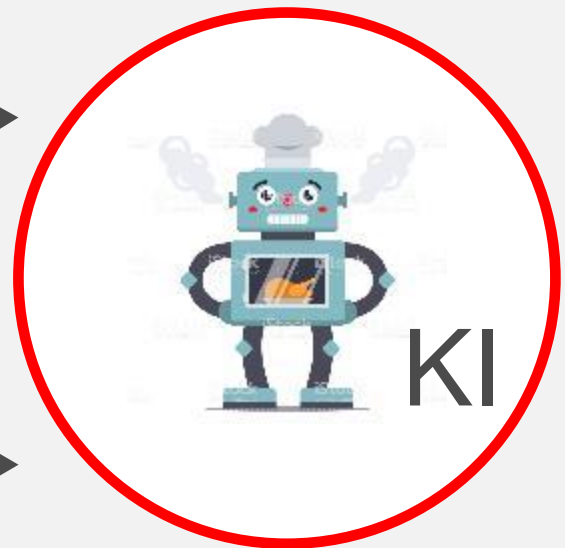
[AI@SCHOOL](#) – [Autonomes Fahren mit dem Calliope mini](#) - Seite 14



Dieses Material steht unter der
Creative-Commons-Lizenz Namensnennung - Weitergabe
unter gleichen Bedingungen 4.0 International.
Um eine Kopie dieser Lizenz zu sehen, besuchen Sie
<http://creativecommons.org/licenses/by-sa/4.0/>

Wie würde die Künstliche Intelligenz backen?

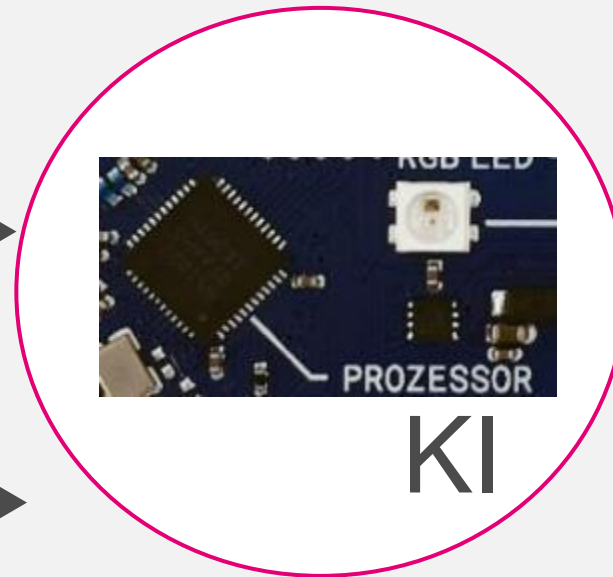
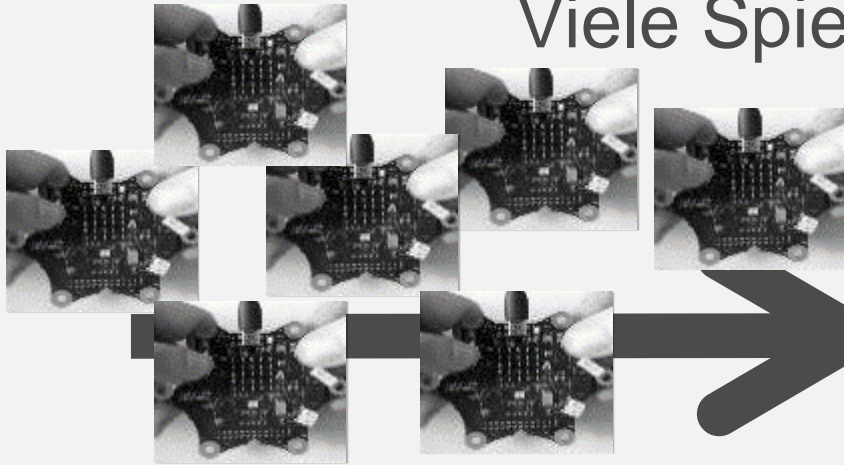
viele Muffins



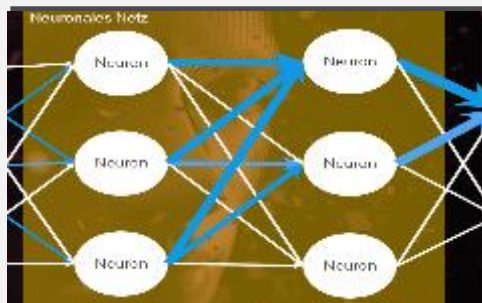
ZUTATEN		ZUBEREITUNG einfache
100 Gramm	Butter	Alles gut verrühren und in d Umluft ca 10-15 Minuten ba beliebte Variation zu Kinder Lebensmittelfarben rot, grün Muffinformen geben und ba
250 Gramm	Mehl	
100 Gramm	Zucker	
1 Prise	Salz	
6 Esslöffel	Milch	
1 Päckchen	Vanillezucker	

Wie funktioniert maschinelles Lernen?

Viele Spieldaten



KI

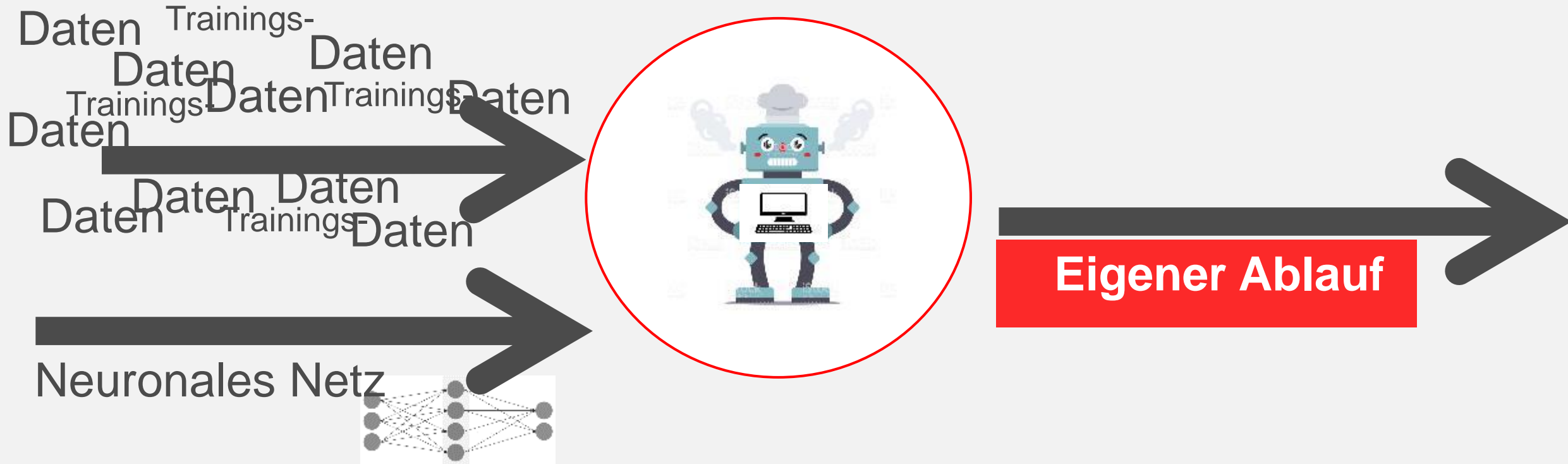


Neuronales Netz



Spielt selber

Wie funktioniert maschinelles Lernen?

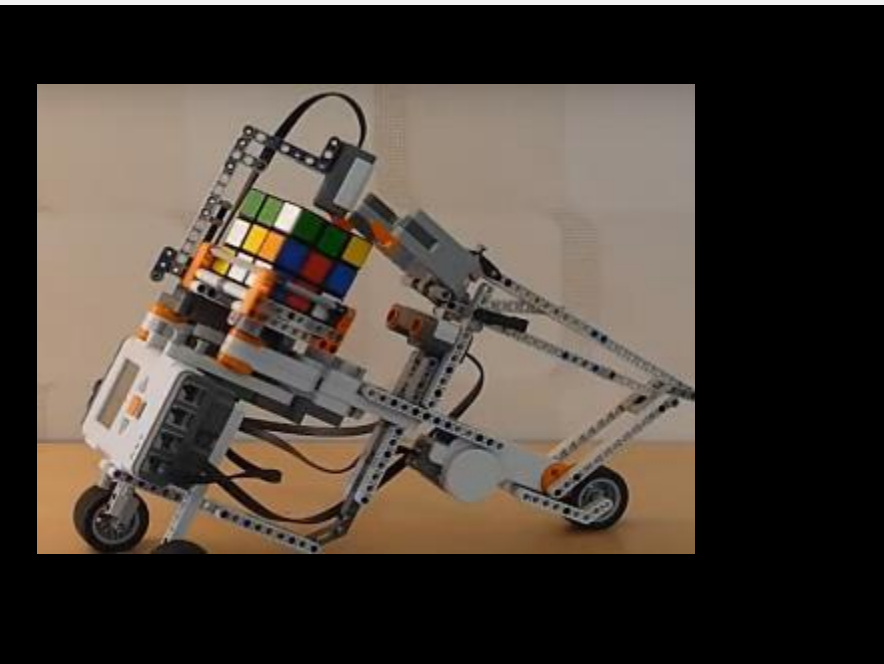
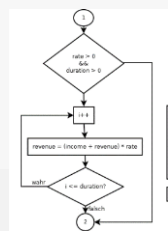


Wo liegt der Unterschied?

Lösen durch Probieren
vom Menschen
(20 - 26 Schritte)

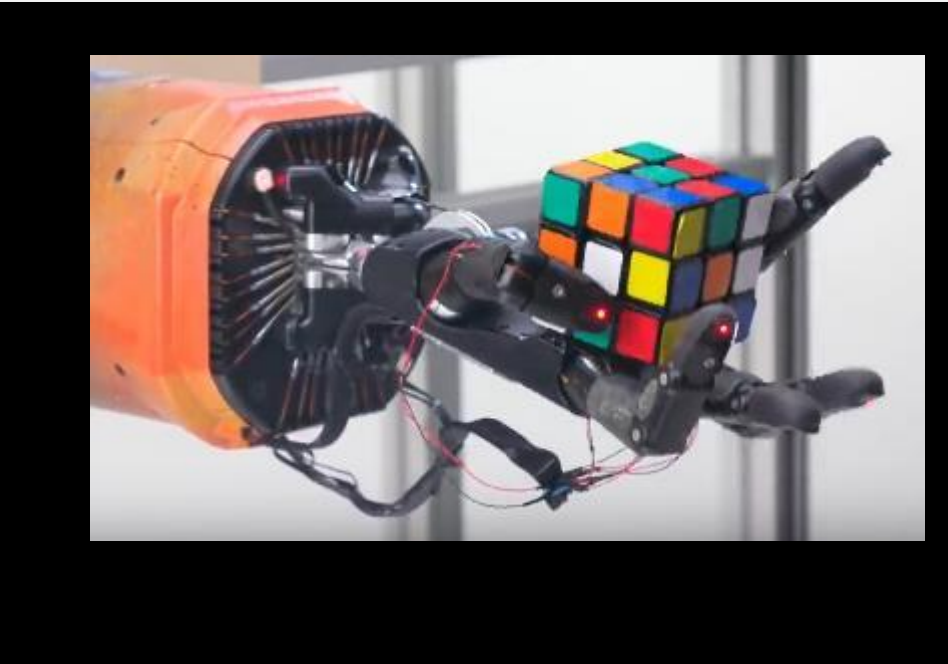
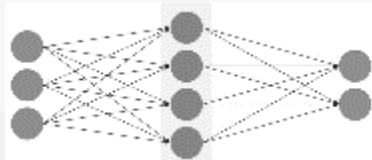


Lösen durch ein fertiges Programm
auf dem Computer
(60 Schritte)



Quelle: <https://www.youtube.com/watch?v=w3f-WyDqOUw>

Lösen durch Trainieren
des Computers
(30 Schritte)

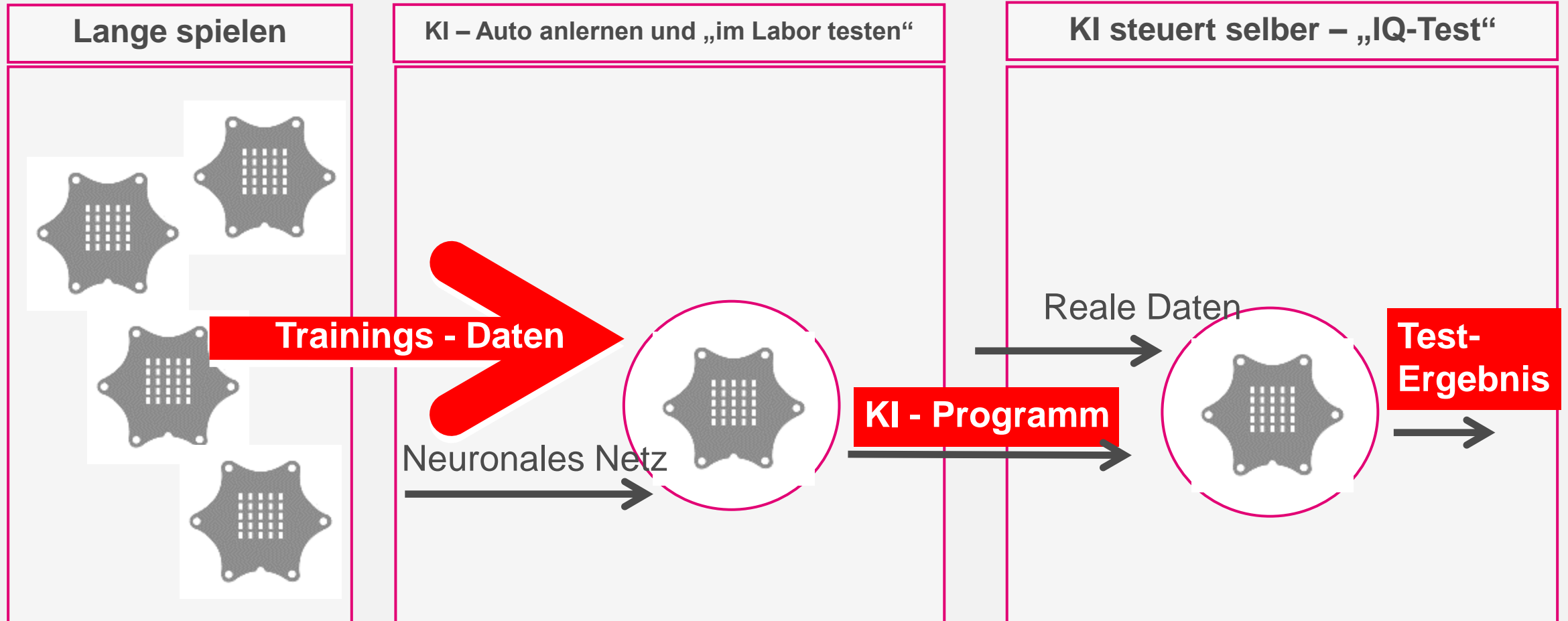


Quelle: <https://youtu.be/kVmp0uGtShk>



Dieses Material steht unter der Creative-Commons-Lizenz Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by-sa/4.0/>

Wie lernen wir nun unseren Calliope an?



Welche Spielsituationen gibt es?

L
I
N
K
S

eigenes Auto	Fahrbahn 1	Fahrbahn 2	Fahrbahn 3	Fahrbahn 4	Fahrbahn 5	Taster A/B/nichts
2	1	0	3	5	2	A

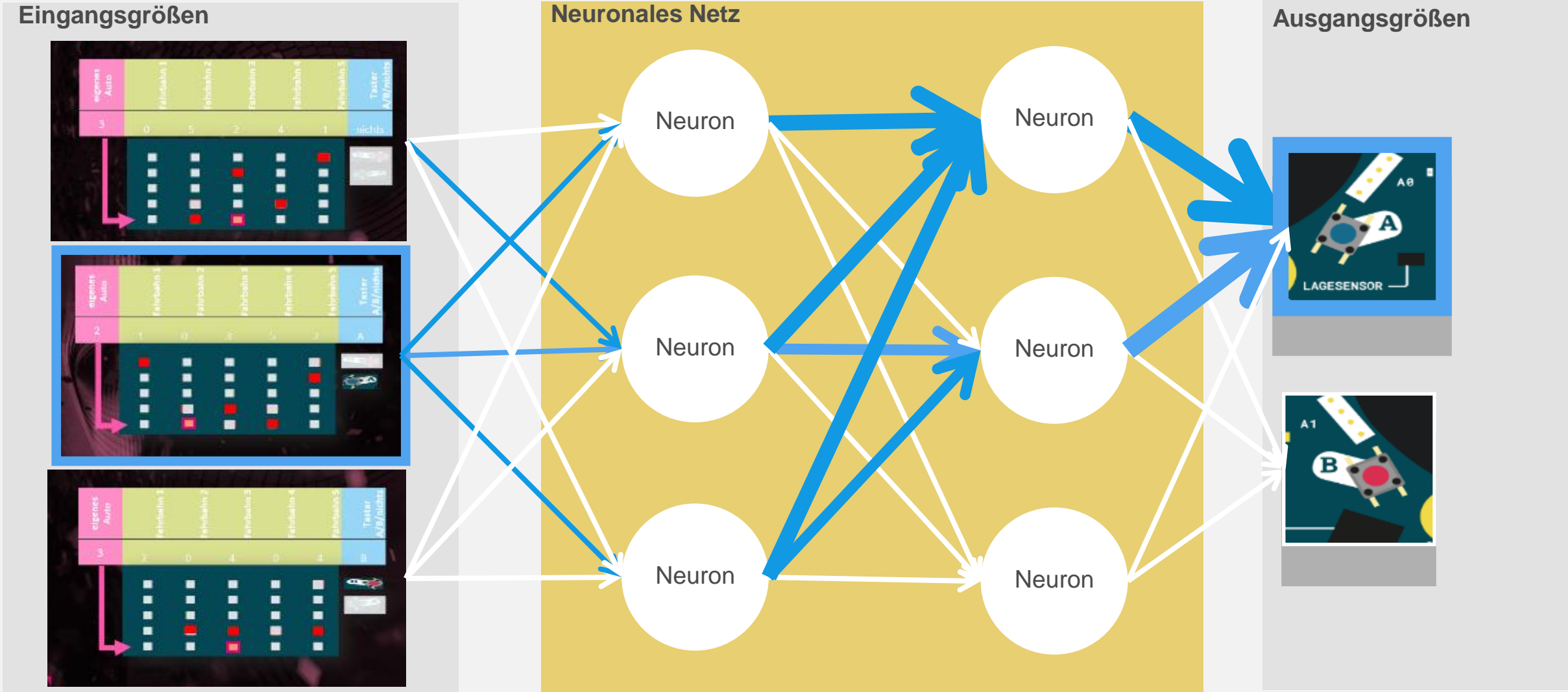
R
E
C
H
T
S

eigenes Auto	Fahrbahn 1	Fahrbahn 2	Fahrbahn 3	Fahrbahn 4	Fahrbahn 5	Taster A/B/nichts
3	2	0	4	0	4	B

N
I
C
H
T
S

eigenes Auto	Fahrbahn 1	Fahrbahn 2	Fahrbahn 3	Fahrbahn 4	Fahrbahn 5	Taster A/B/nichts
3	0	5	2	4	1	nichts

Welche Funktion übernimmt das Neuron?

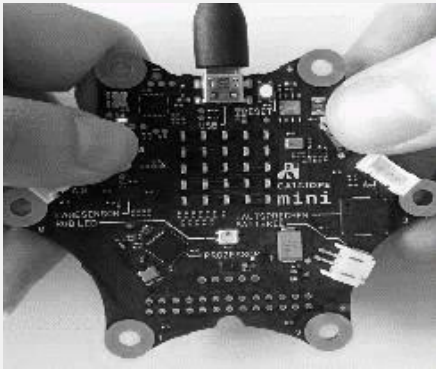


Welches sind die drei Schritte des Anlernens?

1

Lange Spielen um
viele Trainingsdaten
zu erzeugen.

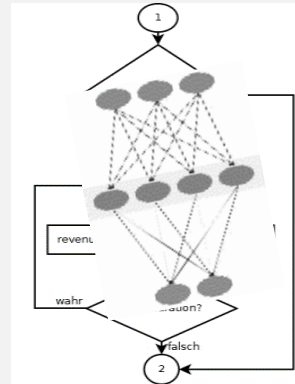
Python-Skript
`ki-datenlogger.py`



2

Eigenes Auto durch
das Neuronale Netz
Anlernen und Testen.

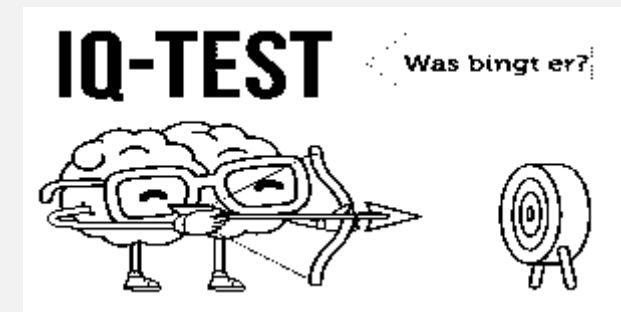
Basisversion: Python-Skript
`ki-trainieren-sklearn.py`
Expertenversion: Orange3
`orange-workflow`



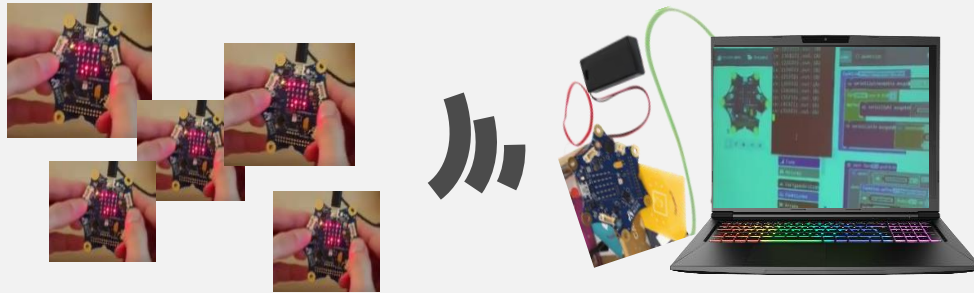
3

Künstlich intelligenter
Calliope übernimmt
die Steuerung und
muss „IQ-Test“
bestehen.

Python-Skript
`ki-rennspiel.py`

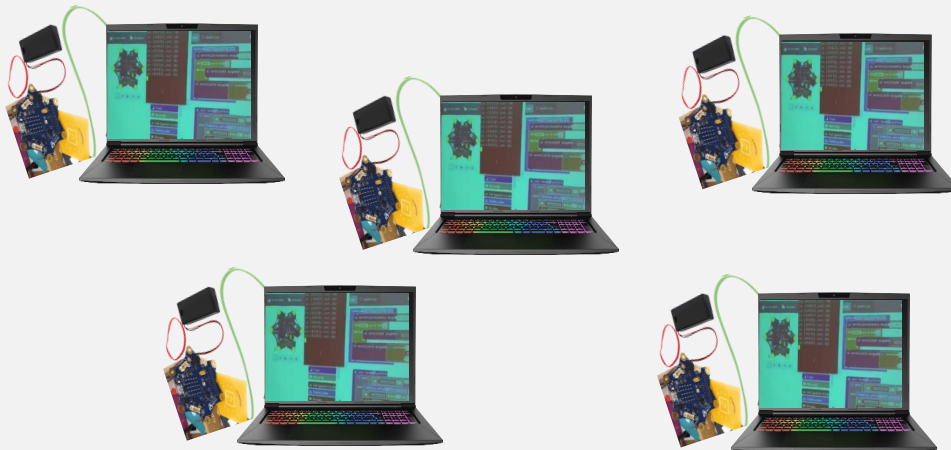


Workshop-Arten



1. Gruppenspiel

- 5-6 Calliopes per Funk an einen Gruppenrechner mit per USB angeschlossenem Datensammler-Calliope
- Softwareinstallation nur auf Gruppenrechnern
- KI-IQ-Test pro Gruppe (Konsolidieren der Daten aller 5-6 Rennspiele)



2. Einzelspiel (Basis- und Expertenversion)

- 1 Calliope an einem Rechner per USB angeschlossen
- Softwareinstallation auf jedem Einzelrechner (jede/r SuS allein)
- KI-IQ-Test auf jedem Einzelrechner (jede/r SuS allein)
- **Basisversion:** Nur KI-Algorithmus “Neuronales Netzwerk”
- **Expertenversion:** Experimentieren mit anderen KI-Algorithmen

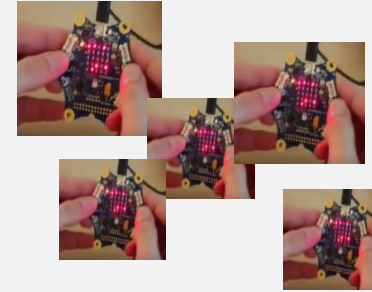
Gruppenspiel

empfohlen: 5-6 Gruppen mit 5-6 Rennspiel-Calliopes → 25-36 SuS

Ausführliche Anleitung für
Gruppenspiel:

<https://github.com/telekom/ki-in-schulen/blob/master/Calliope-Rennspiel/Dokumentation/Nutzeranleitung-Gruppenspielmodus.md>

Gruppe 2



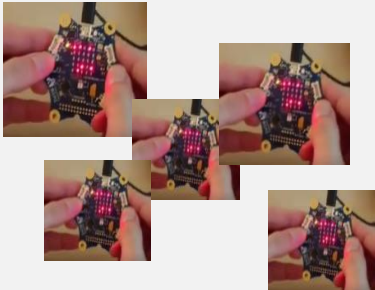
Rennspiel-Calliopes

Calliope-Funk
Kanal **2**



Datensammler-
Calliope per USB an Gruppen-Rechner

Gruppe 1



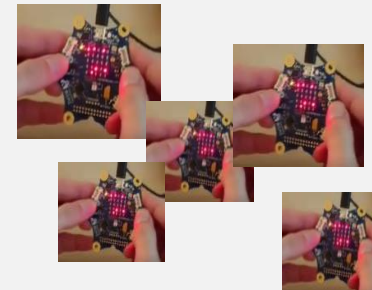
Rennspiel-Calliopes

Calliope-Funk
Kanal **1**



Datensammler-
Calliope per USB an Gruppen-Rechner

Gruppe 3



Rennspiel-Calliopes

Calliope-Funk
Kanal **3**



Datensammler-
Calliope per USB an Gruppen-Rechner



LIFE IS FOR SHARING.

[AI@SCHOOL](https://www.ai@school.de) – [Autonomes Fahren mit dem Calliope mini](https://www.ai@school.de) - Seite 24

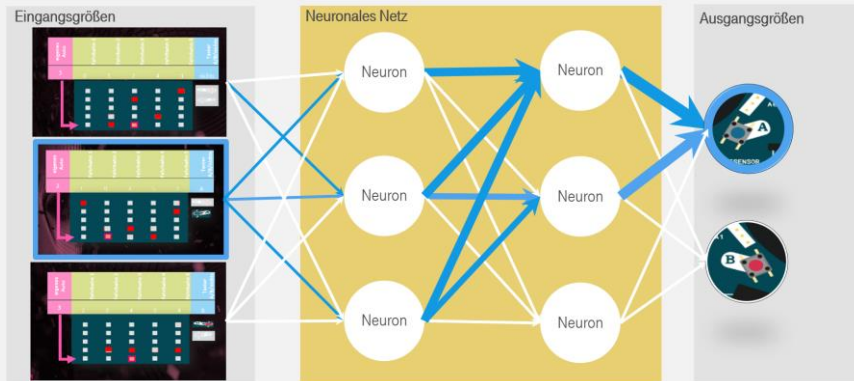


Dieses Material steht unter der
Creative-Commons-Lizenz Namensnennung - Weitergabe
unter gleichen Bedingungen 4.0 International.
Um eine Kopie dieser Lizenz zu sehen, besuchen Sie
<http://creativecommons.org/licenses/by-sa/4.0/>

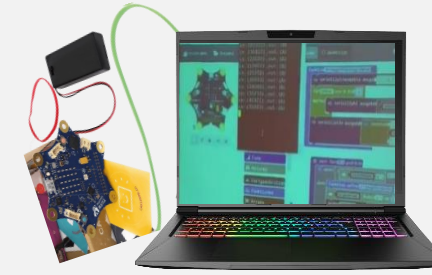
Einzelspiel Basisversion

Ausführliche Anleitung für Einzelspiel:
<https://github.com/telekom/ki-in-schulen/blob/master/Calliope-Rennspiel/Dokumentation/Nutzeranleitung-Einzelspielmodus.md>

Basisversion: Algorithmus „Neuronales Netzwerk“

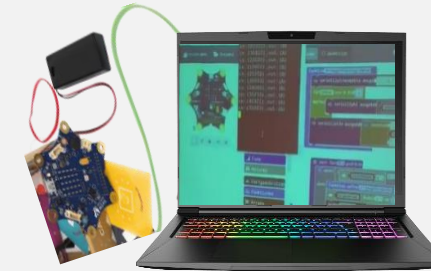


Einzelne/r SuS



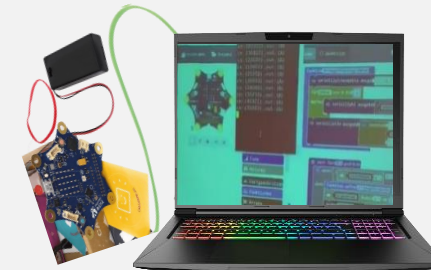
Rennspiel-
Calliope per USB an Einzel-Rechner

Einzelne/r SuS



Rennspiel-
Calliope per USB an Einzel-Rechner

Einzelne/r SuS



Rennspiel-
Calliope per USB an Einzel-Rechner



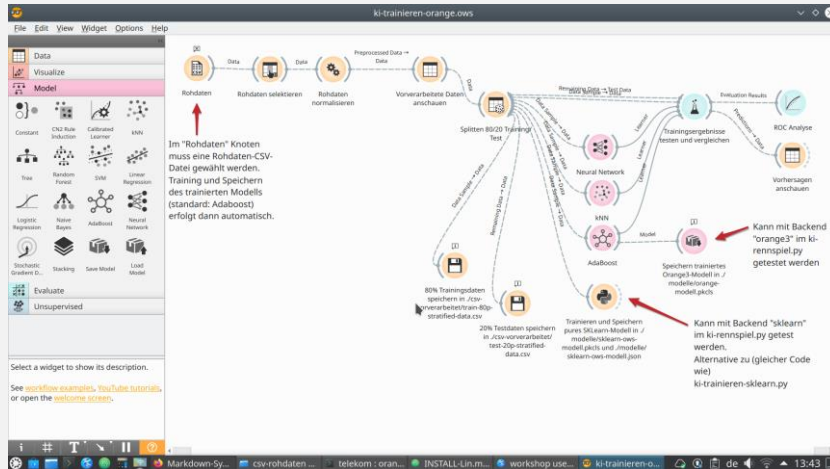
LIFE IS FOR SHARING.



Einzelspiel Expertenversion

Ausführliche Anleitung für Einzelspiel:
<https://github.com/telekom/ki-in-schulen/blob/master/Calliope-Rennspiel/Dokumentation/Nutzeranleitung-Einzelspielmodus-Orange.md>

Basisversion: Vergleich verschiedener KI-Algorithmen

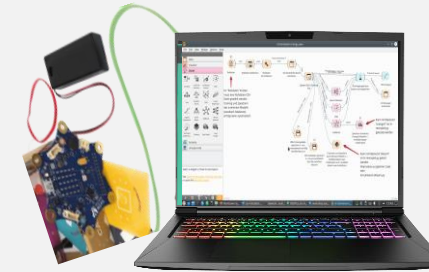


Einzelne/r SuS



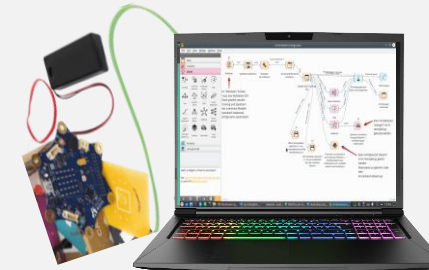
Rennspiel-
Calliope per USB an Einzel-Rechner

Einzelne/r SuS



Rennspiel-
Calliope per USB an Einzel-Rechner

Einzelne/r SuS



Rennspiel-
Calliope per USB an Einzel-Rechner



LIFE IS FOR SHARING.

[AI@SCHOOL](#) – [Autonomes Fahren mit dem Calliope mini](#) - Seite 26



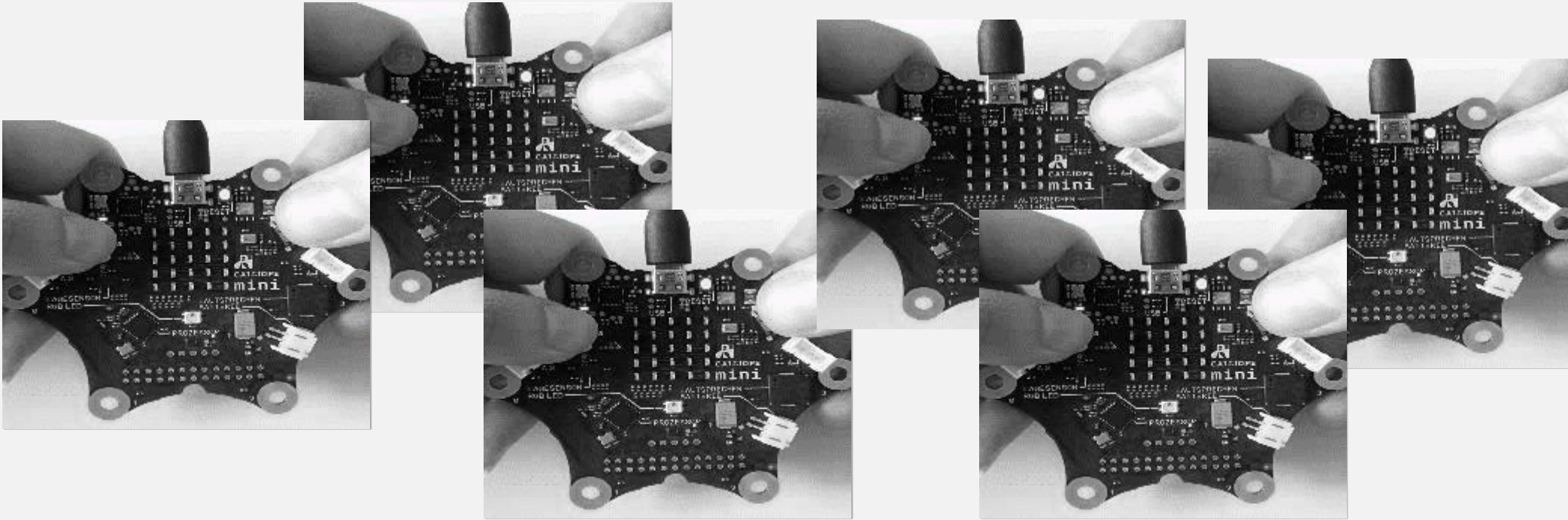
Dieses Material steht unter der Creative-Commons-Lizenz Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by-sa/4.0/>

Trainingsphase

ki-datenlogger.py

1

Lange Spielen um viele Trainingsdaten zu erzeugen.



LIFE IS FOR SHARING.

[AI@SCHOOL](#) – [Autonomes Fahren mit dem Calliope mini](#) - Seite 27



Dieses Material steht unter der Creative-Commons-Lizenz Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by-sa/4.0/>

Wie ist das Spiel aufgebaut? – Trainings-Daten

```

dauerhaft
  ändere output ▼ auf "x"
  wenn gameOn ▼ = wahr ▼ dann
    pausiere (ms) 500 ▼
    wenn output ▼ = "x" dann
      Aufruf erzeugeSnapshot
    +
      Aufruf erzeugeTrainingsdatensatz
    +
      ändere output ▼ auf "x"
  +

```

```

Funktion erzeugeSnapshot
  Array von
    playerCar ▼ x ▼
    car1 ▼ y ▼ ++ 1
    car2 ▼ y ▼ ++ 1
    car3 ▼ y ▼ ++ 1
    car4 ▼ y ▼ ++ 1
    car5 ▼ y ▼ ++ 1
  ändere snapshot ▼ auf
  wenn is car1 ▼ deleted dann
    snapshot ▼ Wert festlegen bei 1 auf 0
  +
  wenn is car2 ▼ deleted dann
    snapshot ▼ Wert festlegen bei 2 auf 0
  +
  wenn is car3 ▼ deleted dann
    snapshot ▼ Wert festlegen bei 3 auf 0
  +
  wenn is car4 ▼ deleted dann
    snapshot ▼ Wert festlegen bei 4 auf 0
  +
  wenn is car5 ▼ deleted dann
    snapshot ▼ Wert festlegen bei 5 auf 0
  +

```

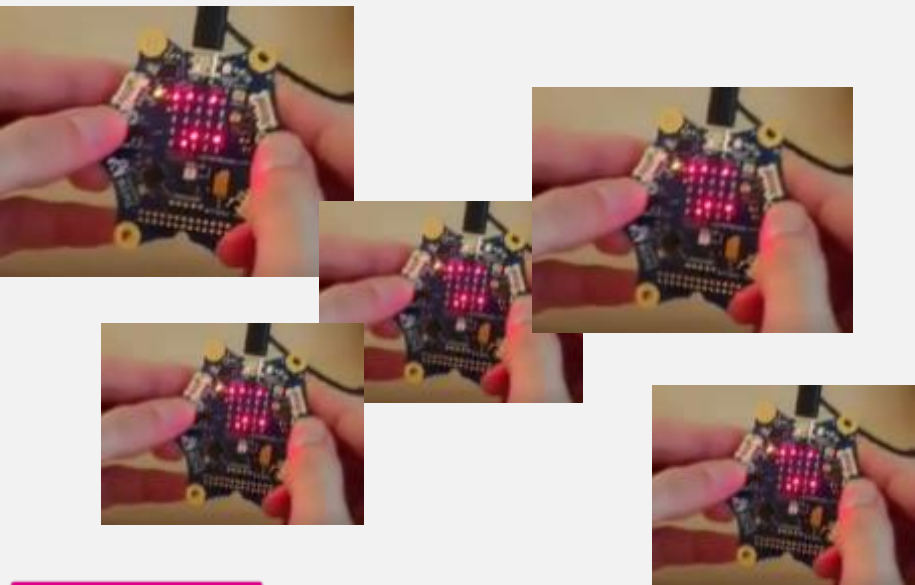
```

Funktion erzeugeTrainingsdatensatz
  für Index von 0 bis 5
  machen
    sende Zahl snapshot ▼ rufe Wert ab bei Index ▼ über Funk
    sende Text " " über Funk
  sende Text output ▼ über Funk

```



Wie ist das Spiel aufgebaut?- Daten sammeln



```
when green flag clicked
  received packet + signal strength + >= 100 then
    when button pressed
      + 1
    else button pressed
      - 1
  else
    when button pressed
      + 1
    else button pressed
      - 1
  when button pressed
    set LED color to blue
  when button pressed
    set LED color to off
```

```
when green flag clicked
  set function to 1
  set variable to 2

function createTrainingDataset
  set output_string to
  for index from 1 to 5
    when button pressed
      set output_string to concatenate output_string, number, index
  set output_string to concatenate output_string, output
  send output_string over USB
```

```
when green flag clicked
  set function to 1
  set variable to 2
  send data output_string

when button pressed
  send data output_string
```

Für Einzelspielmodus wird der Trainingsdatensatz direkt per USB übertragen


```
(ki-calliope-rennspiel-basis) goethe@ubuntu:~/github/telekom/ki-in-schulen/Calliope-Rennspiel/Python$ python ki-datenlogger.py /dev/ttyACM0
Nutze COM-Port: /dev/ttyACM0
1,5,4,3,0,0,x
Nutzdaten erkannt - Übernahme in DataFrame
1,0,5,4,0,0,x
Nutzdaten erkannt - Übernahme in DataFrame
Rennspiel-Funkgruppe1
Nutzdaten erkannt - Übernahme in DataFrame
-----
Nutzdaten erkannt - Übernahme in DataFrame
2,2,1,1,1,1,x
Nutzdaten erkannt - Übernahme in DataFrame
2,3,1,1,1,1,x
Nutzdaten erkannt - Übernahme in DataFrame
2,4,1,1,1,1,x
...
Tastendruck erkannt! Beende Datensammlung und speichere CSV.
Bitte ggf. nochmal Calliope resettet, um Vorgang abzuschließen.
Oder 15 Sekunden warten, dann geschieht dies automatisch.
Rennspiel-Funkgruppe1
Nutzdaten erkannt - Übernahme in DataFrame
  PlayerPos Car1Pos Car2Pos Car3Pos Car4Pos Car5Pos Action
0          1         5         4         3         0         0      x
1          1         0         5         4         0         0      x
2          R         n         s         i         l         F      n
3          -         -         -         -         -         -      -
4          2         2         1         1         1         1      x
5          2         3         1         1         1         1      x
6          2         4         1         1         1         1      x
7          2         5         1         1         1         1      x
8          2         0         1         1         2         1      x
9          2         0         1         1         2         1      A
10         1         0         3         1         4         1      x
11         1         0         4         1         5         1      x
12         1         0         5         2         0         2      x
13         R         n         s         i         l         F      n
Trainingsdaten gespeichert in Datei: ./csv-rohdaten/ki-rennspiel-log-20210316142033.csv
```

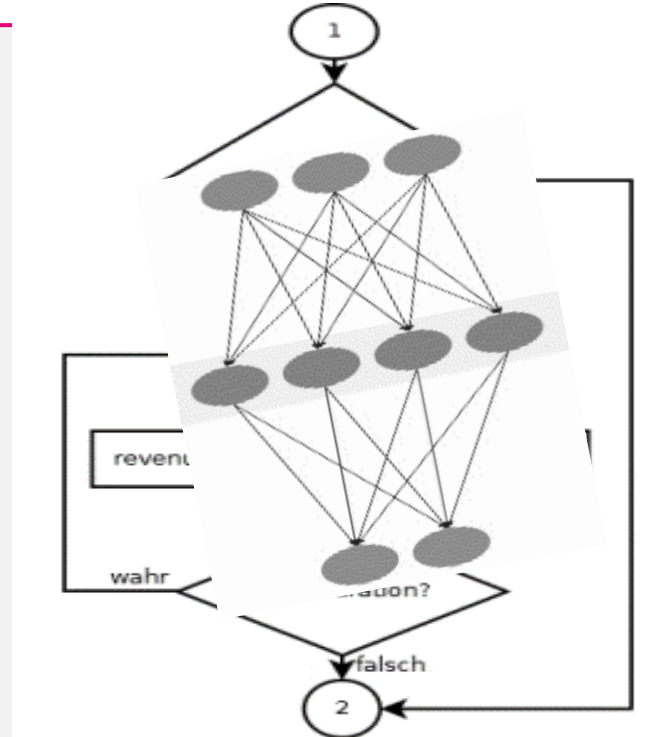
Beliebiger Tastendruck beendet Datensammlung

Trainingsdaten zur Nutzung in Schritt 2
„KI anlernen“ mit **ki-trainieren-sklearn.py**



2

Eigenes Auto durch das Neuronale Netz Anlernen und Testen.



Von **ki-datenlogger.py**
erzeugte Trainingsdaten

```
(ki-calliope-rennspiel-basis) goethe@ubuntu:~/github/telekom/ki-in-schulen/Calliope-Rennspiel/Python$ python ki-trainieren-sklearn.py csv-rohdaten/120minutes-1.csv 7,5,3
Hidden Layers: [7, 5, 3]
Rohdaten Liste der Aktionen (sollten nur x,A,B sein!): ['x' 'B' 'A']
Rohdaten Anzahl x: 6892
Rohdaten Anzahl A: 1092
Rohdaten Anzahl B: 1047
Beginne Trainingsvorgang.
Training beendet.
Erreichte Modellgenauigkeit (Trainingsdaten): 0.8318106312292359
Erreichte Modellgenauigkeit (Testdaten) : 0.8334255672385169
Pickle-Datei des trainierten ML-Modells gespeichert.
Dateiname: ./modelle/sklearn-py-modell-20210316141323.pkcls
JSON-Datei des trainierten ML-Modells gespeichert.
Dateiname: ./modelle/sklearn-py-modell-20210316141323.json
```

Angelerntes neuronales Netzwerk
zur Nutzung in Schritt 3 „IQ-Test“ mit **ki-rennspiel.py**

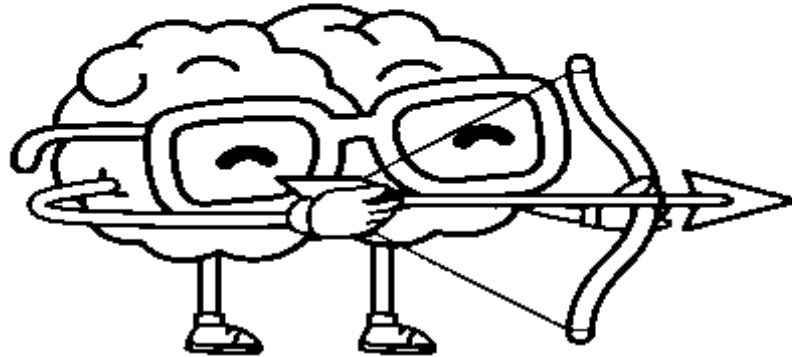


3

Künstlich Intelligente Calliope übernimmt die Steuerung und muss „IQ-Test“ bestehen.

IQ-TEST

Was bingt er?

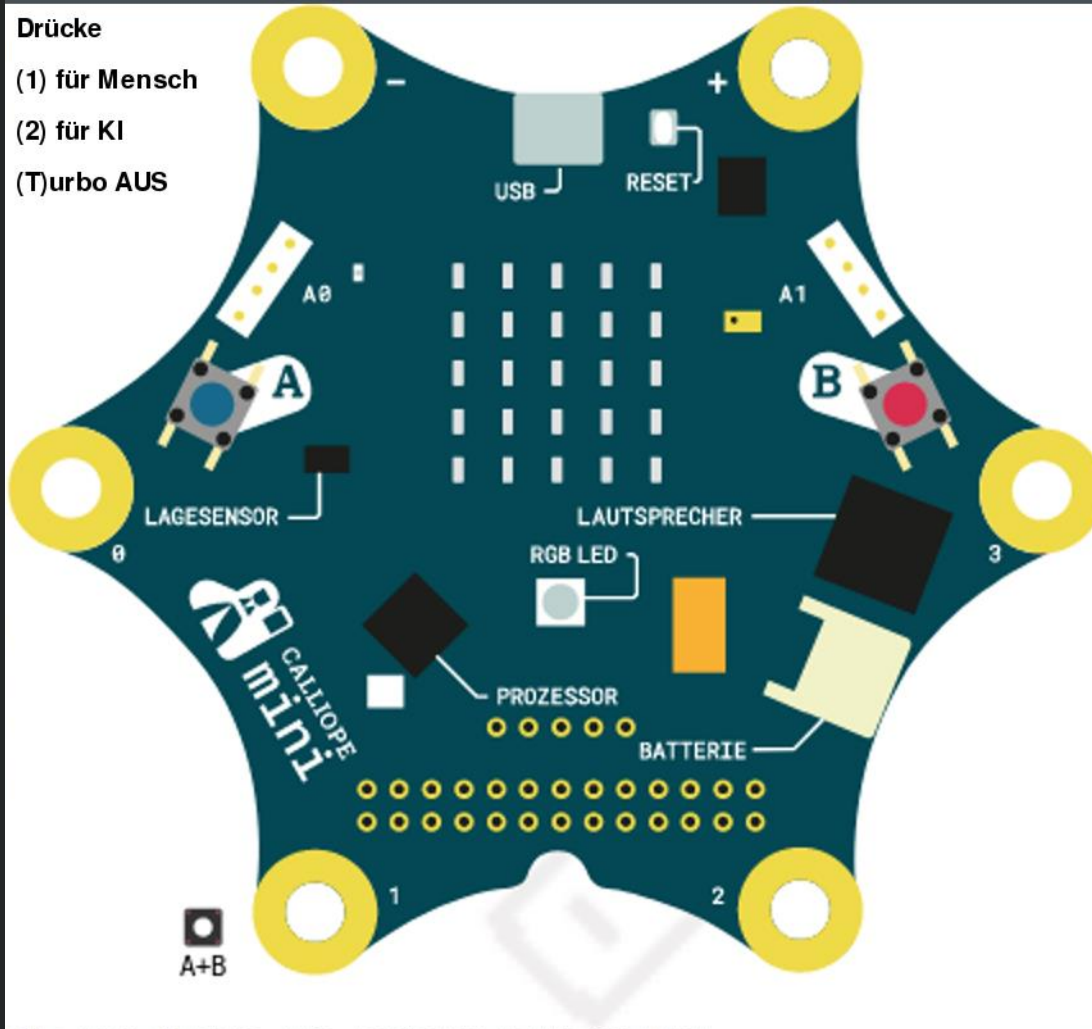


Drücke

(1) für Mensch

(2) für KI

(T)urbo AUS



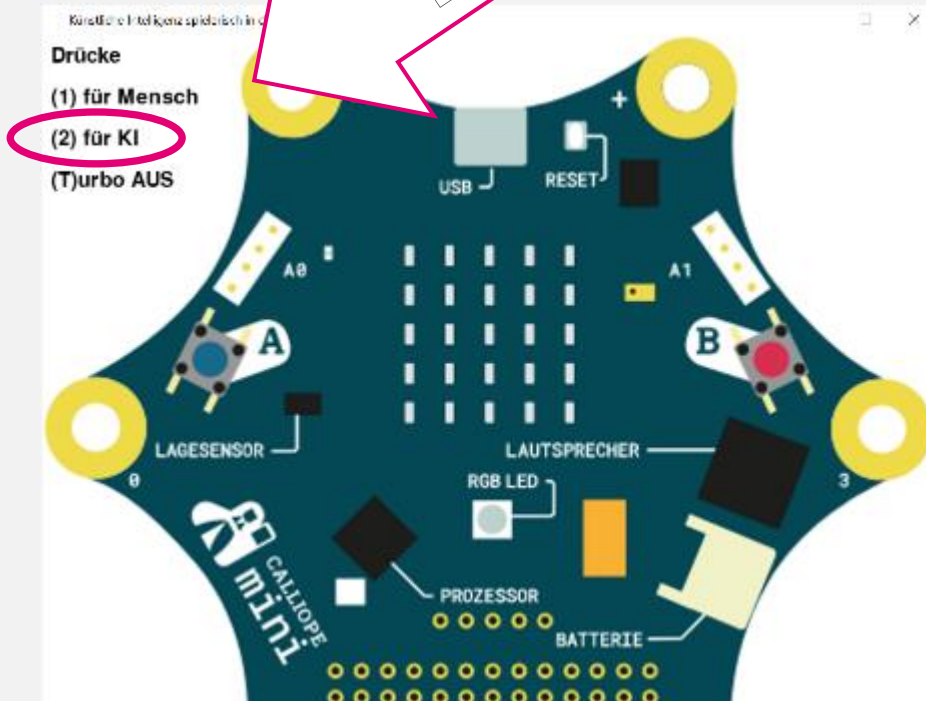
Von **ki-trainieren-sklearn.py**
angelerntes neuronales Netzwerk



```
(ki-calliope-rennspiel-basis) goethe@ubuntu:~/github/telekom/ki-in-schulen/Calliope-Rennspiel/Python$ python ki-rennspiel.py sklearn modelle/sklearn-py-modell-20210316141204.pkcls
pygame 1.9.4
Hello from the pygame community. https://www.pygame.org/contribute.html
Orange nicht verfügbar.
SciKit-Learn Backend, lade ML-Modell modelle/sklearn-py-modell-20210316141204.pkcls
█
```

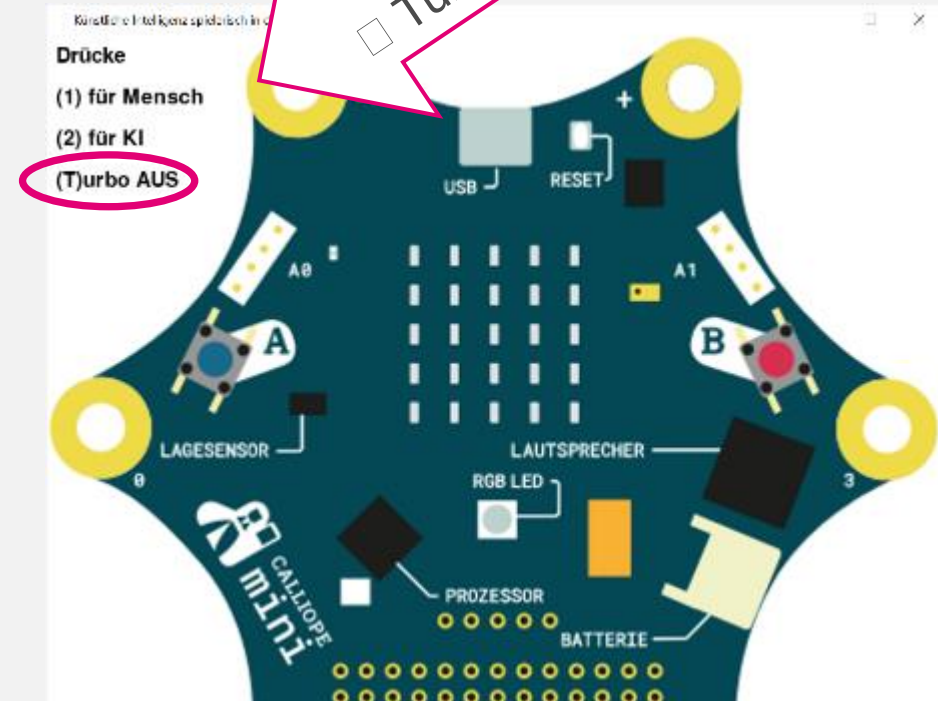

„Beim Fahren
Zuschauen“

Zahl 2 Drücken
☐ KI steuert



„IQ-Test“

Taste T Drücken
☐ Turbo einschalten



Beispiel: 10 Minuten Trainingsdaten



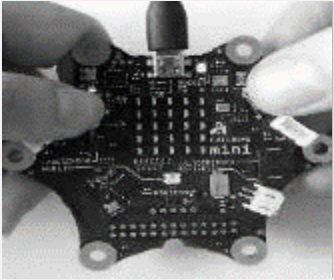
Beispiel: 60 Minuten Trainingsdaten



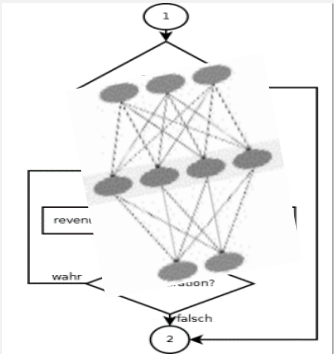
Beispiel: 120 Minuten Trainingsdaten



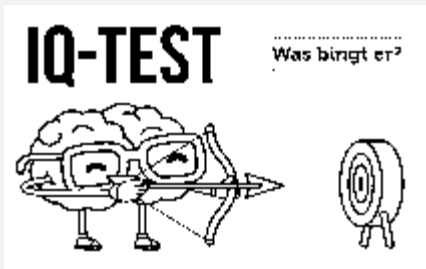
Warum heißt es nun „maschinelles Lernen“?



1. Wir haben viele Trainings-Daten durch Spielen erzeugt.



2. Wir haben nun den KI-Calliope angelernt und getestet.



3. Wir haben den KI-Calliope alleine spielen lassen und wissen nun, wie gut er ist.