

KÜNSTLICHE INTELLIGENZ (KI) IN SCHULEN

WISSEN ZU KÜNSTLICHER INTELLIGENZ SPIELERISCH IN DIE SCHULEN BRINGEN

FERENC HECHLER, MIRKO JELINEK, CHRISTIAN SCHILLER, DIRK WOLTERS



ERLEBEN, WAS VERBINDET.

PROJEKTMATERIAL UND LIZENZ

- Die neueste Version des Projektmaterials liegt immer auf GitHub:
 - <https://github.com/telekom/ki-in-schulen> (Schulen in Deutschland)
 - ??? <https://github.com/telekom/ai-in-schools> (placeholder for international rollout)
 - ??? Projekt klonen in andere europäische Sprachen?
- Lizenz für Dokumentation und Code
 - tbd
- Primärkontakt für das Projekt:
 - Christian Schiller, Senior IT & Enterprise Architect, Deutsche Telekom IT GmbH, Landgrabenweg 151, 53227 Bonn
 - Emailkontakt: christian.schiller@telekom.de
 - GitHub Kontakt: <https://github.com/c-a-schiller>



<Lizenzlogo>



ERLEBEN, WAS VERBINDET.

EINLEITUNG

- Das Deutsche Telekom IT@School Projekt „**Künstliche Intelligenz (KI) in Schulen**“ hat ein Workshopformat entwickelt, was Grundkenntnisse zum Thema KI spielerisch vermittelt
 - Geeignet für Mittel- und Oberstufe
 - Workshopformat kann flexibel angepasst werden an die Situation an einer Schule (Kenntnisstand Lehrkräfte und SuS)
- Das Projekt wurde initial für den Lerncomputer **Calliope** mit **Microsoft Makecode** entwickelt
- Das Projekt wird Open Source auf GitHub gestellt, um:
 - eine Verbreitung zu ermöglichen und via GitHub „Pull-Requests“ Mitarbeit zu fördern
 - eine Portierung zu anderen Hardware- und Software-Plattformen zu ermöglichen (bspw. Calliope mit OpenRoberta)



INHALT DIESES DOKUMENTS

- Einführung und Überblick
- Softwareanforderungen
 - für Workshopvariante „Calliope Mini & Microsoft Makecode“
- Details zum Workshopformat und Ideen zur Einbettung in ein Curriculum



EINFÜHRUNG UND ÜBERBLICK

WARUM IST ES WICHTIG, SCHON FRÜHZEITIG IM THEMA „KÜNSTLICHE INTELLIGENZ“ (KI) ZU BILDEN?

- KI wird im IT-Bereich eine der **Basistechnologien** des 21. Jahrhunderts sein
- Basis von KI ist das sogenannte **Maschinelle Lernen**
- KI ist fundamental unterschiedlich zu klassischer Programmierung
 - Stochastisch statt deterministisch – ein **Paradigmenwechsel**
- KI ist schon im **Alltag** angekommen, sowohl **positiv** als auch **negativ**
 - Sprach-, Übersetzungs-, Bildverarbeitungs-Assistenten auf Smartphones
 - Robotik: Haushaltsroboter, Autonomes Fahren
 - Filterblasen & Meinungslenkung durch KI-Newsfeeds
 - Deep Fake News

Grundlagenwissen in Künstlicher Intelligenz
stärkt die Technologie- und Medienkompetenz!



WORKSHOPIDEE - ÜBERBLICK

- **Lernziel**

- Kinder lernen auf spielerische Art und Weise:
 - was “künstliche Intelligenz” ist: Einsatz von Algorithmen des **maschinellen Lernens**
 - den **gravierenden Unterschied** zwischen klassischer Programmierung von Computern und deren Programmierung mittels maschinellen Lernens

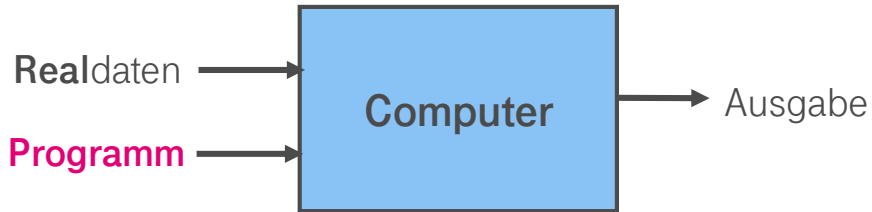
- **KI-Workshop für Kinder**

- Kinder bringen in einem Workshop einem “autonomen Auto” das selbständige (autonome) Fahren bei, indem sie einen Calliope “künstlich intelligent” machen, sodass er autonom Hindernissen auf der Straße ausweichen kann
- Dies zur Erhöhung der Motivation durch „Gamification“:
 - Kinder erzeugen Trainingsdaten für die KI mittels eines Autorennspiels auf der LED-Matrix des Calliope
 - Teams können gebildet werden, die gegeneinander antreten, das beste autonome Auto zu entwickeln
 - Wer das beste autonome Auto “programmiert” (trainiert), gewinnt



DER PARADIGMENWECHSEL BEIM MASCHINELLEN LERNEN

Klassische Programmierung

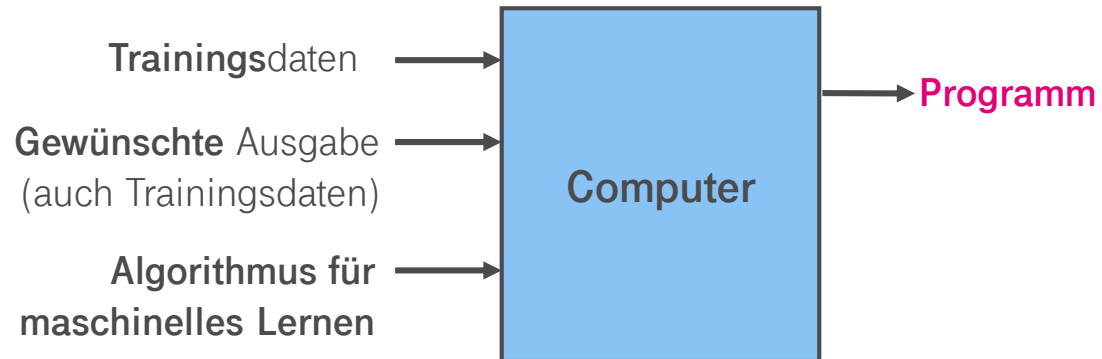


„Software 1.0“

100% deterministisches Verhalten
(vorhersagbares Verhalten)



Programmierung mittels maschinellen Lernens



„Software 2.0“

„100-X %“ → stochastisches Verhalten
(nur in einer **Bandbreite** von Wahrscheinlichkeiten
vorhersagbares Verhalten)

Inspiration: [Jason Brownlee](#)



ERLEBEN, WAS VERBINDET.

– öffentlich – Christian Schiller – KI@Calliope: Autonomes Fahren – 16.12.2019

NATÜRLICHE INTELLIGENZ

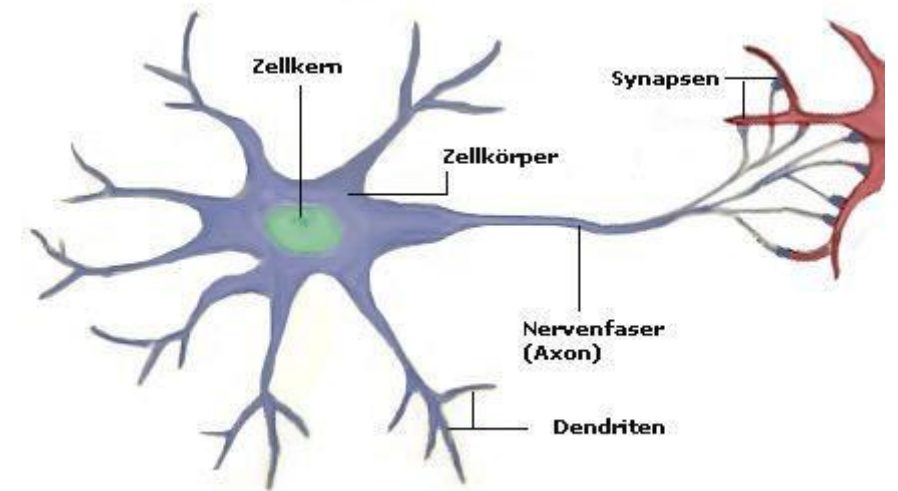
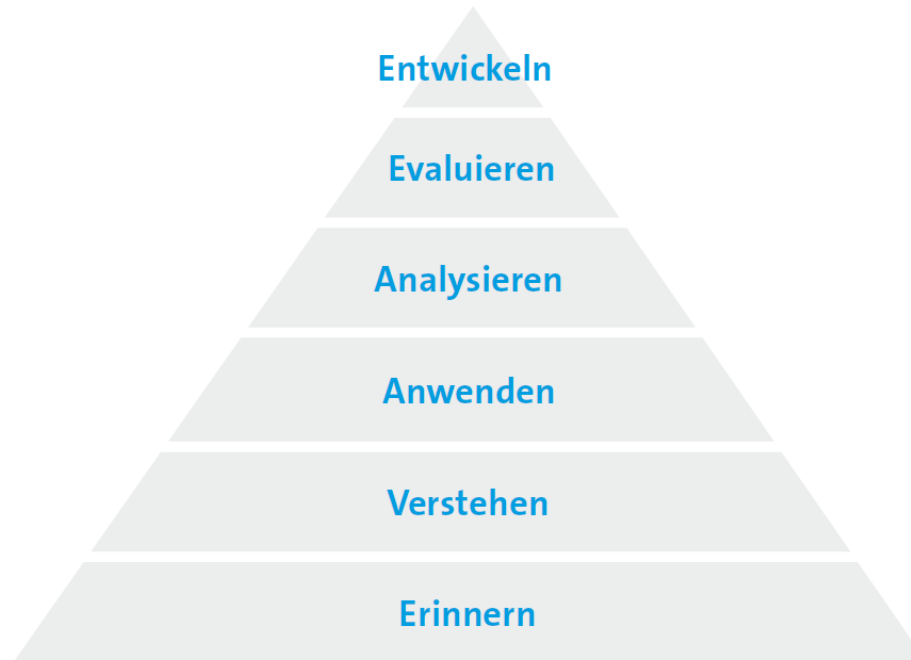
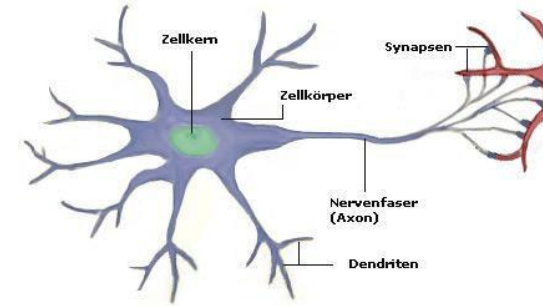


Abbildung 2: Hierarchische Klassifikation kognitiver Fähigkeiten²⁸

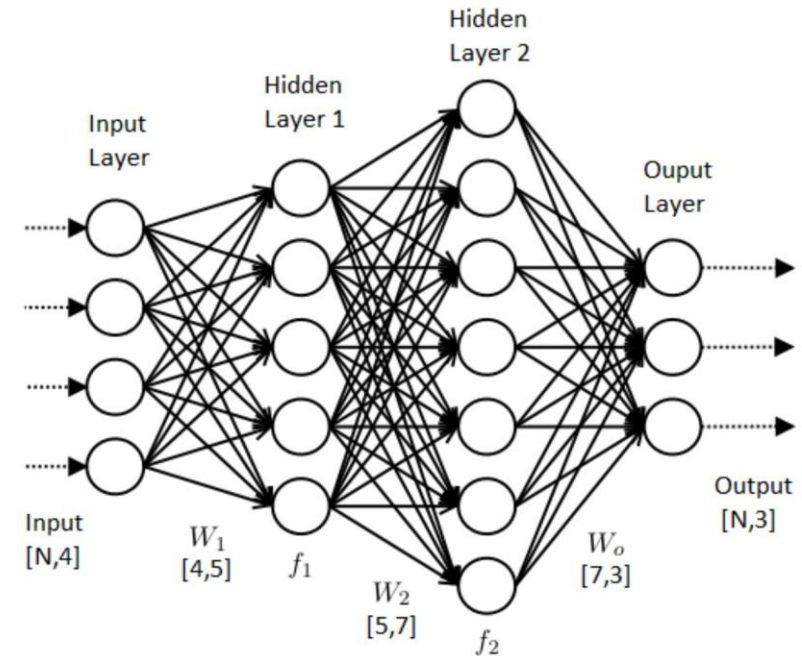
KÜNSTLICHE „INTELLIGENZ“

- Es gibt sehr **viele Varianten** des maschinellen Lernens, jede hat ihre Vor- und Nachteile
- Obwohl mathematisch relativ komplex, sind **künstliche neuronale Netzwerke (KNN)**, die von der Biologie des menschlichen Hirns inspiriert sind, die auch für Laien am eingängigsten zu erklärende Variante
- Stand heute sind KNN jedoch nur **sehr krude** Nachbildungen des menschlichen Hirns.
- Die resultierenden „Intelligenzen“ sind eher als „**Fachidioten**“ zu bezeichnen, die für ein klar umgrenztes Problemfeld genauso gut oder gar besser sein können als ein Mensch, aber noch lange nicht an die holistische menschliche Intelligenz heranreichen

Echte Neuronen



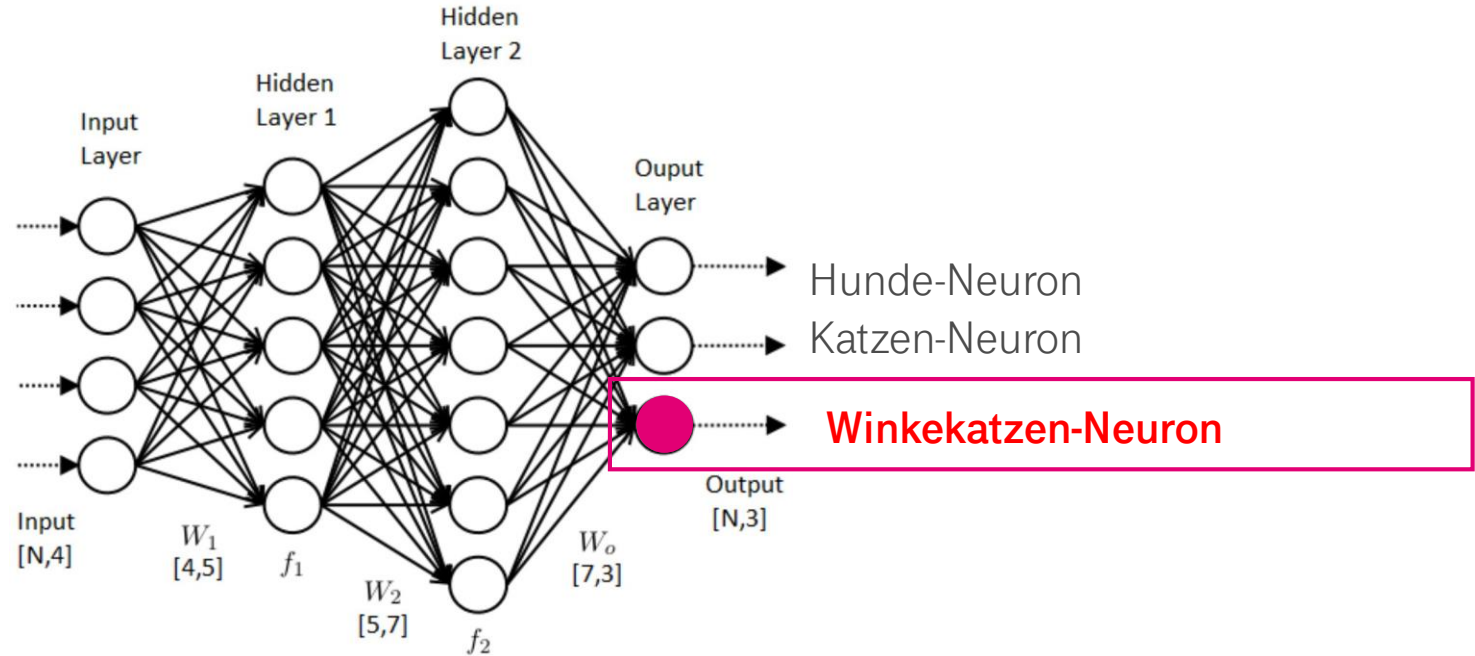
Künstliche Neuronen



KNN – GRUNDSÄTZLICHE FUNKTIONSWEISE



Bildquelle

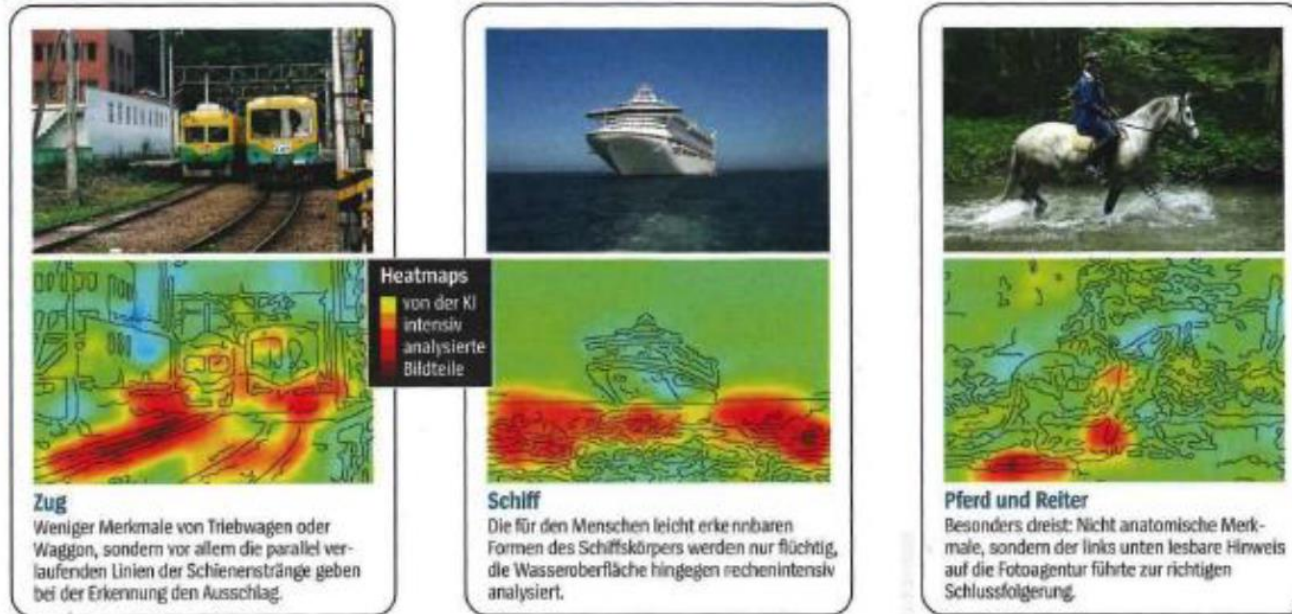


Bei einem angelernten/trainierten KNN sind die Gewichte der Neuronenverbindungen (W_x) so gesetzt (=angelernt/trainiert) worden, dass bei einem eingegebenen Winkekatzen-Bild Idealerweise immer das „Winkekatzen-Neuron“ aktiviert wird

KI „DENKT“ GANZ ANDERS ALS DER MENSCH

(NOCH?) NICHT VERGLEICHBAR MIT NATÜRLICHER INTELLIGENZ

Quelle: Der Spiegel 33/10.08.2019, p. 101



- **Rot** hervorgehoben ist, woran eine einzelne angelernte KI Züge/Schiffe/Pferde+Reiter erkennt (*eine anders angelernte KI würde es anders erkennen!*)

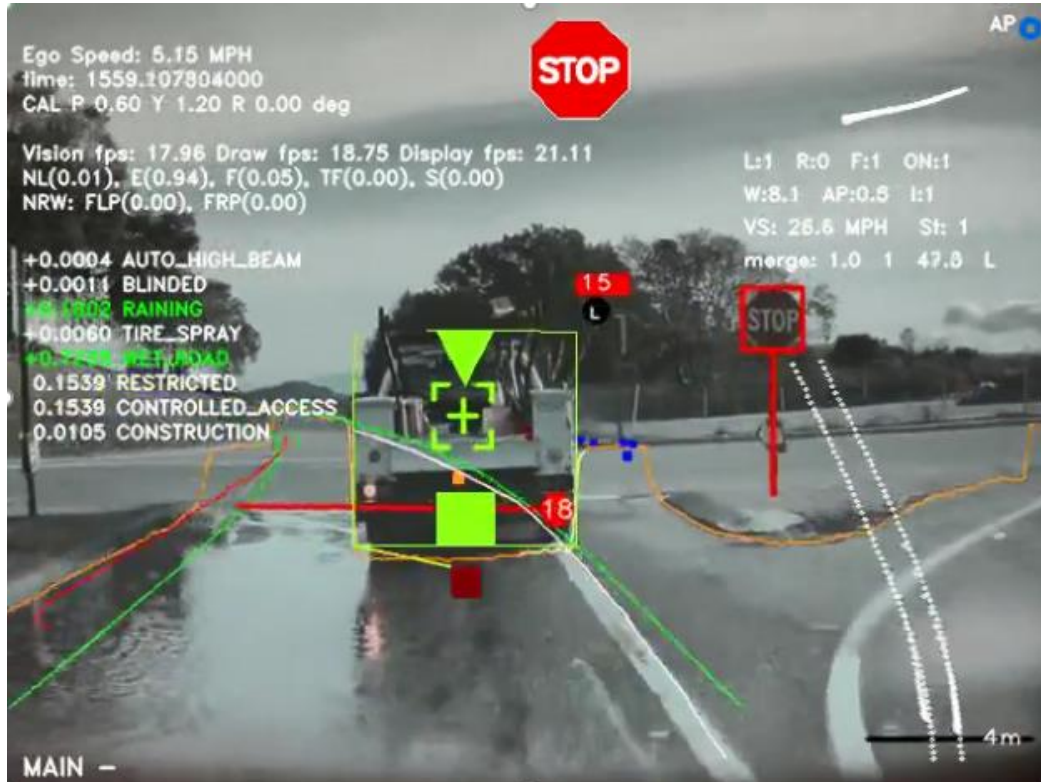
Züge → Gleise

Schiffe → Wasserflächen

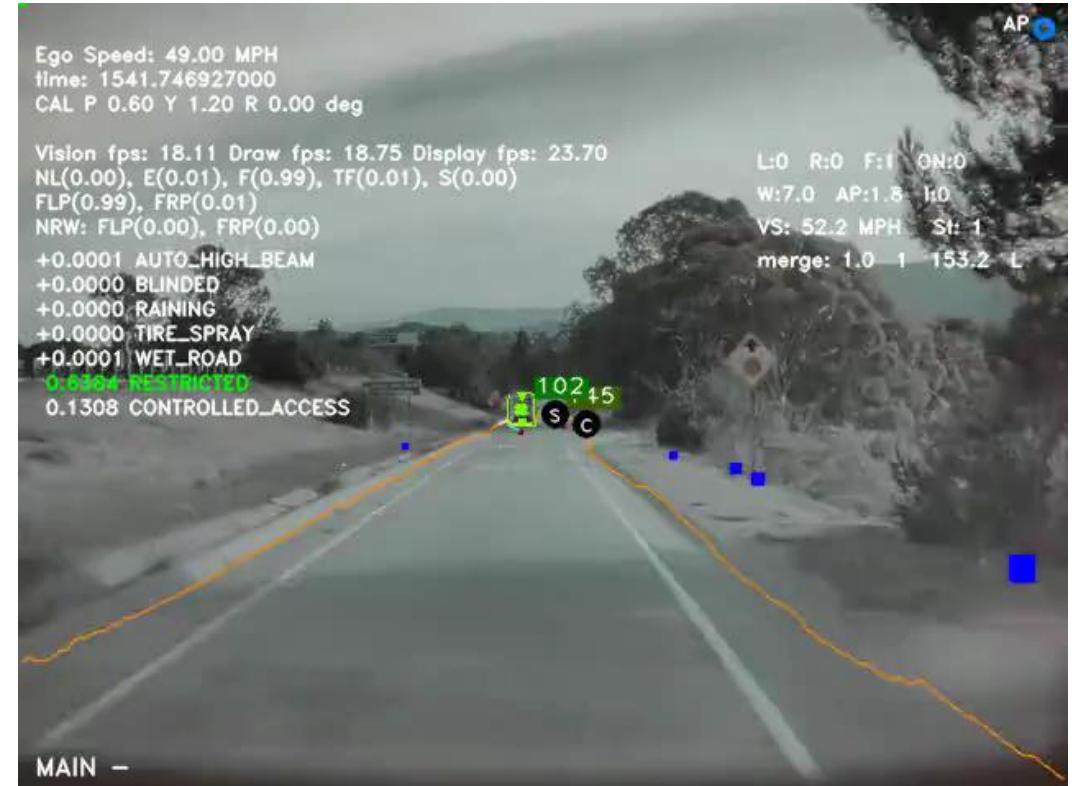
Pferde+Reiter → Logo der Fotoagentur, die die Bilder bereit gestellt hat

AUTONOMES FAHREN IN DER ECHTEN WELT

So “sieht” ein Tesla Autopilot die Verkehrssituation: <https://www.tesla.com/autopilotAI>



Standbild (für PDF)



Eingebettetes Video (nur Powerpoint)

DES WORKSHOPS KERN

Wir wollen auf dem Calliope Mini
selbst ein autonomes Auto entwickeln.

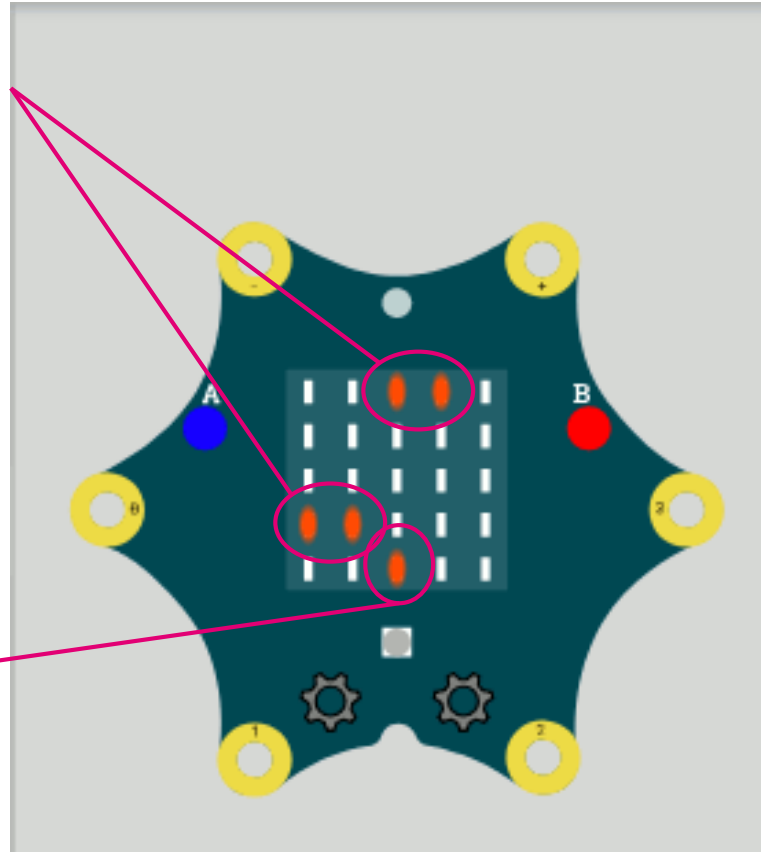
Wie geht das?



AUTONOMES FAHREN AUF DEM CALLIOPE MINI

Andere Verkehrsteilnehmer

Das autonome Auto



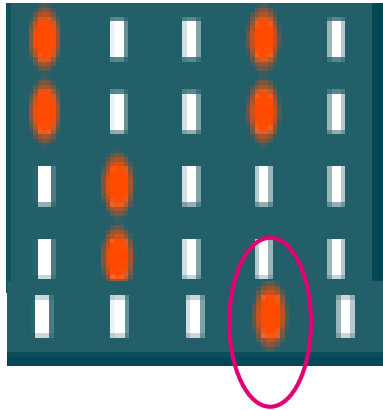
Die Straße

VERGLEICH ECHE WELT MIT UNSEREM WORKSHOP

„WOW“-EFFEKT FÜR DIE SCHÜLER

- Das „autonome Calliope-Auto“ hat im Vergleich mit der Realität nur wenig Komplexität zu bewältigen (links)
- Aber die Grundprinzipien der KI – maschinelles Lernen – sind genau die gleichen!
- Die KI-Ingenieure bei den Autokonzernen machen prinzipiell nichts anderes, müssen dabei „nur“ eine „etwas“ höhere Komplexität bewältigen (siehe Foto rechts)

Das „Calliope-Auto“
sieht das hier vor sich:

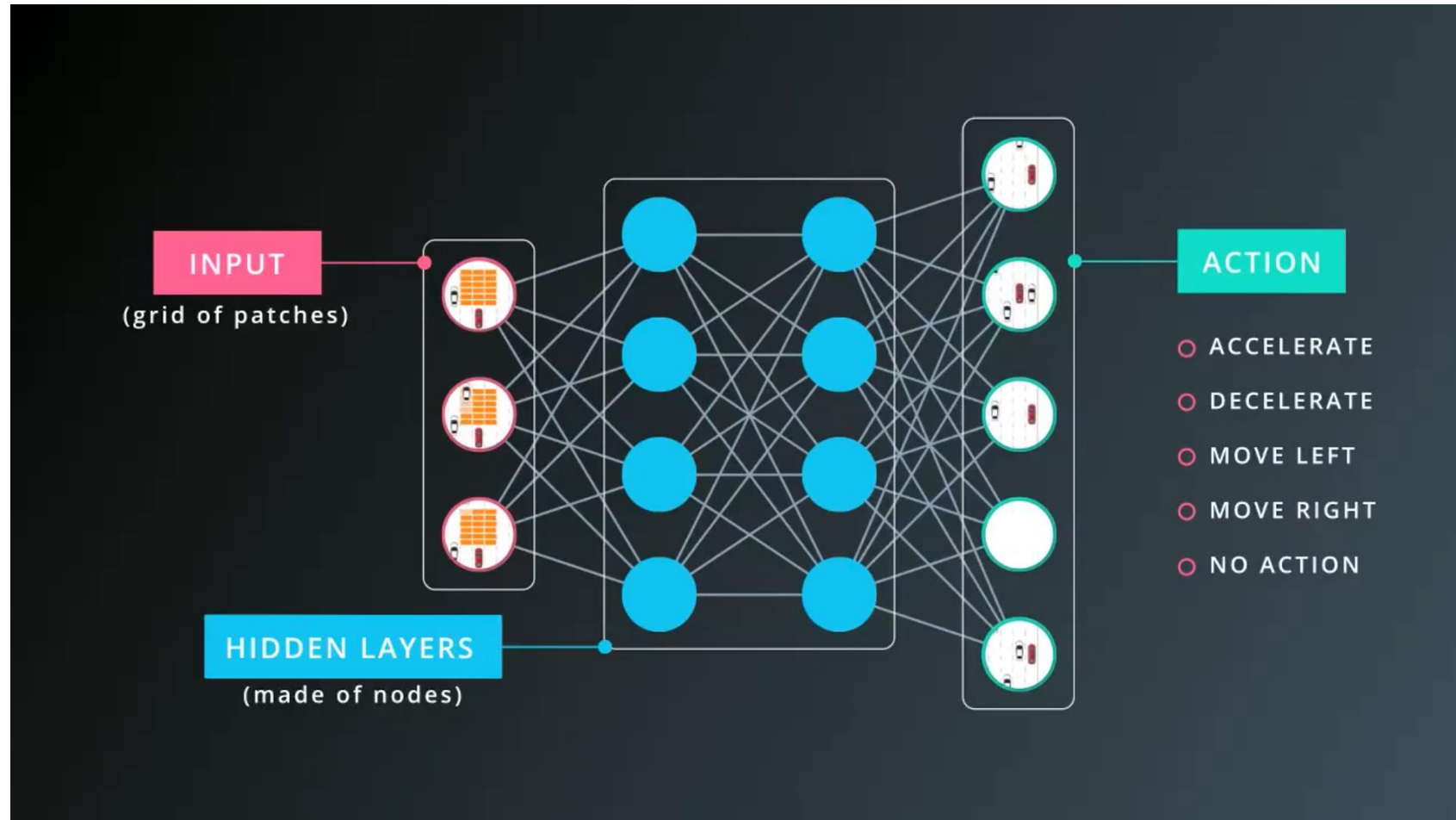


Ein echtes Auto
sieht sowas vor sich:



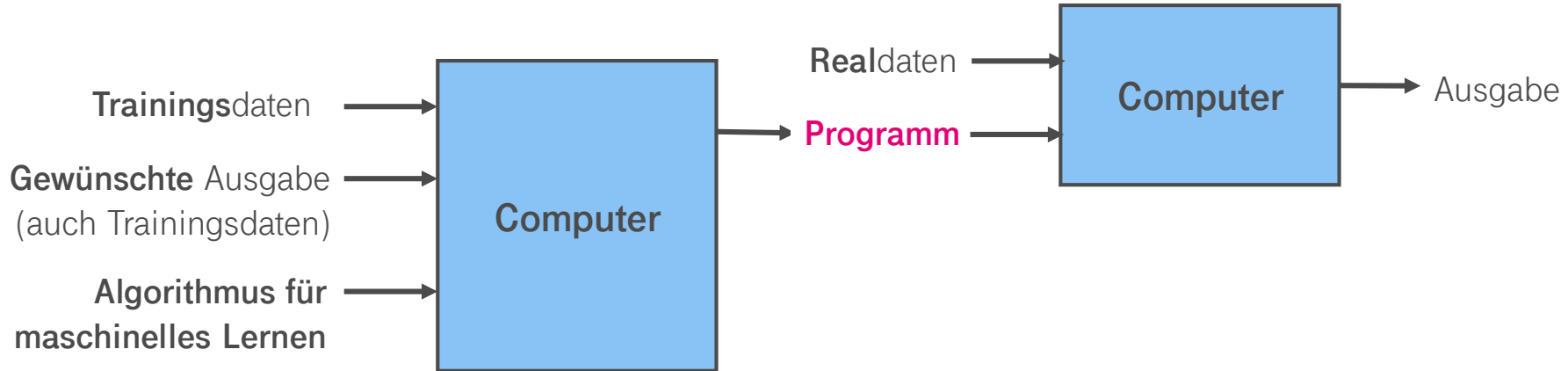
Dieser Pixel hier ist das
steuerbare „Calliope-Auto“

VERARBEITUNG DES CALLIOPE-AUTORENNSPIELS MIT KÜNSTLICHEM NEURONALEN NETZ



DER LERNPROZESS FÜR DAS „AUTONOME CALLIOPE-AUTO“

VERKNÜPFUNG VON KLASSISCHER PROGRAMMIERUNG MIT MASCHINELLEM LERNEN



Trainingsdaten und gewünschte Ausgabe erzeugen

Programm erstellen mittels Algorithmus für maschinelles Lernen

Angelerntes Programm ausführen

Anwendungs-
Perspektive

Testfahrten durchführen

Das künstliche Gehirn des autonomen Autos anlernen

Künstliches Gehirn übernimmt die Steuerung und muss „IQ-Test“ bestehen

Technische
Perspektive

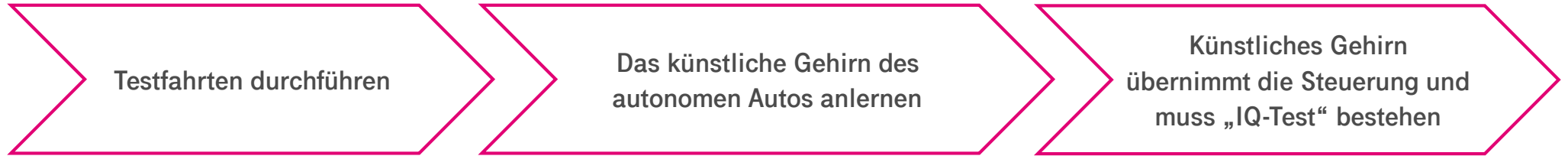
Trainingsdaten sammeln

ML Modell trainieren

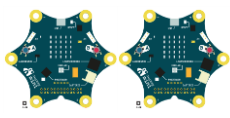
ML Modell anwenden und testen



UNSER PROJEKT UNTERSTÜTZT MEHRERE VARIANTEN FÜR DEN LERNPROZESS DES „AUTONOMEN CALLIOPE-AUTOS“

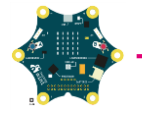


Variante 1A & 1B


Jeder Schüler individuell

ODER


Pooling Trainingsdaten von Schülergruppen

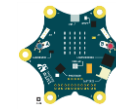

(Daten-Sammler)

CSV



JSON

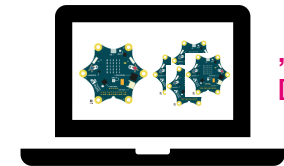
1A



„IQ-Test“ direkt auf Calliope

Pickle

1B



„IQ-Test“ auf Desktop-Rechner

Desktop-Rechner für Anlernen künstliche neuronale Netzwerke und auch andere KI-Algorithmen

Variante 2

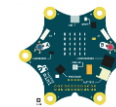

Jeder Schüler individuell

ODER


Pooling Trainingsdaten von Schülergruppen


Daten-Sammler

Zusätzlich zur Datensammlung Lernt der Calliope ein eigenes künstliches neuronales Netzwerk an

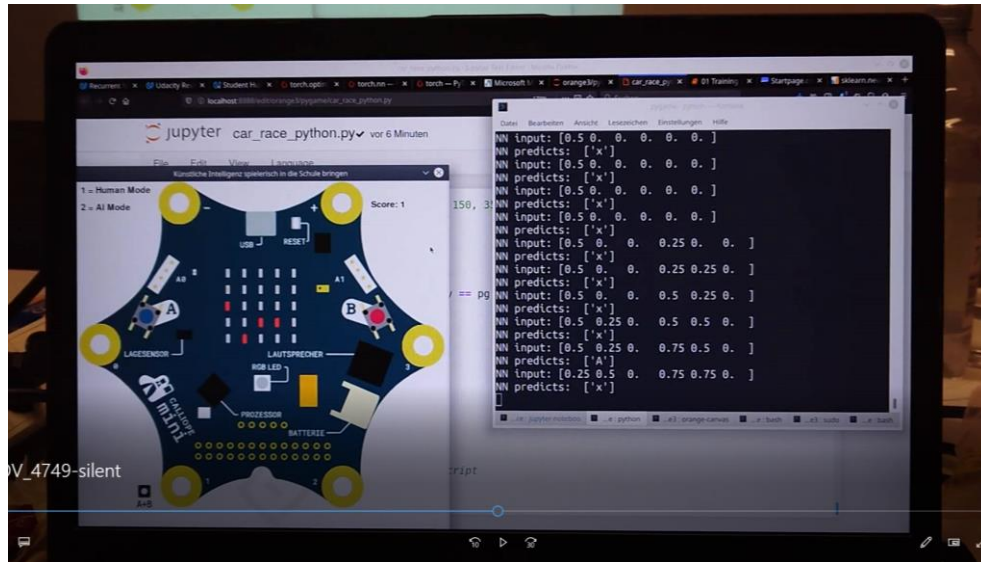


„IQ-Test“ direkt auf Calliope

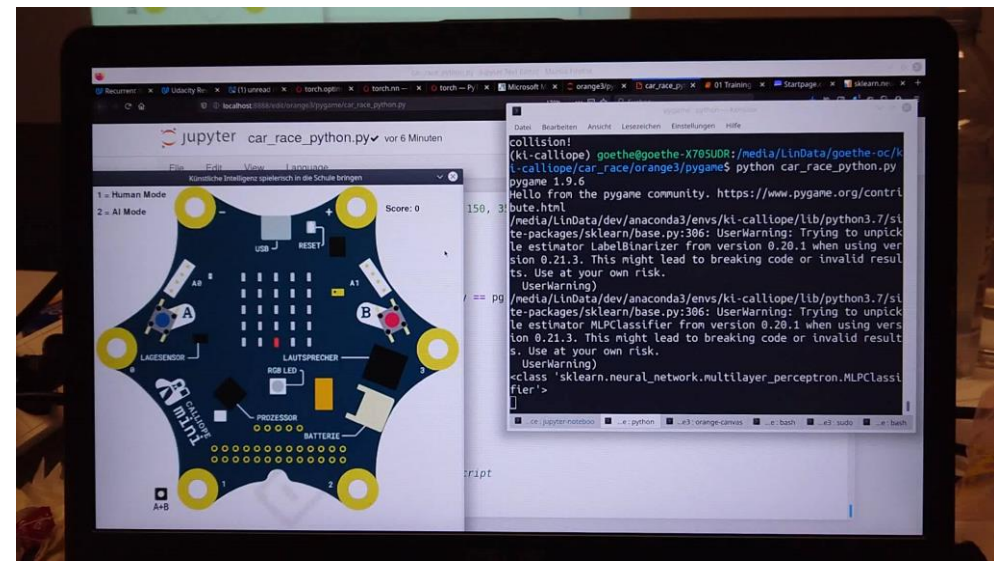
VIDEO ZU VARIANTE 1B

Video ansehen auf:

- LinkedIn: https://www.linkedin.com/posts/christianschiller_ai-artificialintelligence-ki-activity-6613046241179774976-SUba
- Twitter: https://twitter.com/c_a_schiller/status/1207283815552557056



Standbild (für PDF)



Eingebettetes Video (nur Powerpoint)

SOFTWAREANFORDERUNGEN FÜR WORKSHOPVARIANTE „CALLIOPE MINI UND MICROSOFT MAKECODE“

REALISIERUNGSSTATUS STAND 03/2020

Variante	Nötige Hardware	Basissoftware für Nutzung der Projektsoftware	Trainingsdaten sammeln	ML Modell trainieren	ML Modell anwenden und testen	Status
1A	Calliope, Desktop	Calliope Microsoft Makecode; Python: - Scikit-Learn	Calliope + Desktop	Desktop	Desktop, Calliope	6.3.: - Übertragung eines JSON-NN zum Calliope erfolgreich - Offen: Übernahme der Steuerung des Calliope-Autos durch das übernommene Netz
1B	Calliope, Desktop	Calliope Microsoft Makecode; Python: - Scikit-Learn, - PyGame, - Orange (optional)	Calliope + Desktop	Desktop	Desktop	6.3.: - Einsatzbereit! - offen: Nutzbarkeitsverbesserungen (bspw. zentrales Nutzerinterface statt Skriptnutzung)
2	Calliope	Calliope Microsoft Makecode;	Calliope	Calliope	Calliope	in Arbeit



LISTE DER BASIS- UND PROJEKTSOFTWARE

- Auf den folgenden Folien ist der Inhalt des GitHub des Projektes beschrieben
 - Dokumentation
 - Softwaregrundlagen und Projektdateien für **Projektvariante 1A** („IQ-Test auf Calliope“)
 - Softwaregrundlagen und Projektdateien für **Projektvariante 1B** („IQ-Test auf Desktop-PC“)



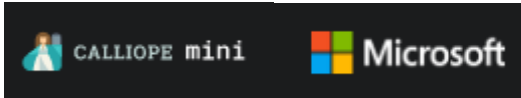
PROJEKTDATEN - DOKUMENTATION

- 1 20200312 KI-Calliope Projektbrief_Draft v5.pptx
- 2 Konzept_.Künstliche_Intelligenz_leicht_erklärt_v01.docx
- 3 Künstliche_Intelligenz_leicht_erklärt_20022 8.pptx

- 1 Gesamtüberblick über Projekt (dieses Dokument)
- 3 **Projektvariante 1B** - SuS-geeigneter Foliensatz mit Workshopablauf und Nutzungsanleitung – für die Workshops
- 2 **Projektvariante 1B** – Erläuterung zu SuS-geeignetem Foliensatz



SOFTWAREGRUNDLAGEN FÜR VARIANTE 1A



- **Calliope Makecode**
 - (Stand 11.3.2020: Betaversion) <https://makecode.calliope.cc/beta>
Programmierungsumgebung für grafische Calliope Programmierung
- **C**
 - ### TODO FERI
 - .
 - .
 - .
 - .
- **Python 3.7.6**
 - **Scikit-Learn 0.22.1**
<https://scikit-learn.org/stable/>
Bibliothek für maschinelles Lernen (Erzeugen von KI-Modellen)



VARIANTE 1A – PROJEKTDATEN FÜR MAKECODE

- ### TODO gemeinsam
 - Rennspiel (car_race_makecode)
 - Brauchen wir für Variante 1A einen Fork/neue Version von car_race_makecode_v15
 - Oder schaffen wir eine integrierte Version, die für beide Varianten nutzbar ist



VARIANTE 1A – PROJEKTDATEN FÜR C

- **### TODO FERI**

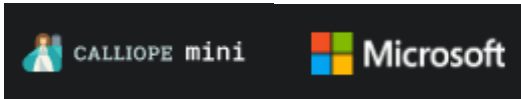
- Ist es möglich, dass zur Integration der calliope makecode NN extensions von Dir Feri
 - ein subfolder eines github Projektes angegeben wird für den Import
 - oder brauchen/wollen wir auf dem telekom github ein zweites projekt (ein klon von deinem projekt auf deinem github bzw. deinem projekt auf codesharewb)



VARIANTE 1A – PROJEKTDATEN FÜR PYTHON

<code>car_race.py</code>	für Variante 1A nicht benötigt	Python-Rennspiel für den „IQ-Test“ (Auswertung Leistung des trainierten Autos)
<code>ki-calliope-conda-environment.yml</code>		} Konfigurationsdateien für Anaconda und pip zur Installation notwendiger Python-Pakete
<code>ki-calliope-pip-requirements.txt</code>		
<code>ki-collect.py</code>		Datenlogger zur Annahme von Trainingsdaten von einem oder mehreren Calliopes
<code>ki-train.py</code>		Trainieren eines KI-Modells direkt mit Rohdaten auf Shell (Anaconda/Python Prompt)
<code>Pipeline1B-Betal.ows</code>	für Variante 1A nicht benötigt	Trainieren eines KI-Modells unter Nutzung der grafischen Workflow-Umgebung Orange
<code>sprite.png</code>	für Variante 1A nicht benötigt	} Python-Rennspiel Grafiken – nicht löschen!
<code>surface.png</code>		
<code>csv-preprocessed</code>	für Variante 1A nicht benötigt	Leerverzeichnis für Orange Workflow – nicht löschen!
<code>csv-raw</code>		} 1. Bereits gesammelte Beispiel-Rohdaten vom Calliope-Rennspiel 2. Vom Datenlogger (ki-collect.py) genutztes Verzeichnis
<code>10minutes-1.csv</code>		
<code>120minutes-1.csv</code>		
<code>20minutes-1.csv</code>		
<code>30minutes-1.csv</code>		
<code>40minutes-1.csv</code>		
<code>50minutes-1.csv</code>		
<code>60minutes-1.csv</code>		
<code>models</code>		Leerverzeichnis für trainierte KI-Modelle – nicht löschen!

SOFTWAREGRUNDLAGEN FÜR VARIANTE 1B



- **Calliope Makecode**

- (Stand 11.3.2020: Betaversion erforderlich) <https://makecode.calliope.cc/beta>
Programmierungsumgebung für grafische Calliope Programmierung

- **Python 3.7.6**

- **Scikit-Learn 0.22.1**

<https://scikit-learn.org/stable/>

Bibliothek für maschinelles Lernen (Erzeugen von KI-Modellen)

- **Pygame 1.9.6**

<https://www.pygame.org/news>

Bibliothek für Python Spiele – für Auswerten der Ergebnisse

- **Orange 3.24.1** (optional für Fortgeschrittene)

<http://orange.biolab.si/>

Plattform für Datenanalyse und maschinelles Lernen



VARIANTE 1B – PROJEKTDATEN FÜR CALLIOPE MAKECODE

<code>car_race_makecode_v15.hex</code>	}	Calliope Mini Autorennspiel zum Sammeln von Trainingsdaten
<code>car_race_makecode_v15.js</code>		
<code>Datensammler_makecode_v03.hex</code>	}	Calliope Datensammler
<code>Datensammler_makecode_v03.js</code>		

- Makecode-Weboberfläche: <https://makecode.calliope.cc/beta>
- Die .hex Dateien wurden durch den Compiler der o.g. Makecode-Weboberfläche schon ins Binärformat des Calliope Mini kompiliert und können direkt auf einen per USB angeschlossenen Calliope kopiert werden, oder in der o.g. Makecode-Weboberfläche als Projekt importiert werden.
- Die .js Dateien können in der o.g. Makecode-Weboberfläche per copy/paste in ein leeres neues Projekt (Ansicht: JavaScript) übernommen werden



VARIANTE 1B – PROJEKTDATEN FÜR PYTHON

<code>car_race.py</code>	Python-Rennspiel für den „IQ-Test“ (Auswertung Leistung des trainierten Autos)
<code>ki-calliope-conda-environment.yml</code>	} Konfigurationsdateien für Anaconda und pip zur Installation notwendiger Python-Pakete
<code>ki-calliope-pip-requirements.txt</code>	
<code>ki-collect.py</code>	Datenlogger zur Annahme von Trainingsdaten von einem oder mehreren Calliopes
<code>ki-train.py</code>	Trainieren eines KI-Modells direkt mit Rohdaten auf Shell (Anaconda/Python Prompt)
<code>Pipeline1B-Betal.ows</code>	Trainieren eines KI-Modells unter Nutzung der grafischen Workflow-Umgebung Orange
<code>sprite.png</code>	} Python-Rennspiel Grafiken – nicht löschen!
<code>surface.png</code>	

`csv-preprocessed` — Leerverzeichnis für Orange Workflow – **nicht löschen!**

`csv-raw`

`10minutes-1.csv`
`120minutes-1.csv`
`20minutes-1.csv`
`30minutes-1.csv`
`40minutes-1.csv`
`50minutes-1.csv`
`60minutes-1.csv`

1. Bereits gesammelte Beispiel-Rohdaten vom Calliope-Rennspiel
2. Vom Datenlogger (ki-collect.py) genutztes Verzeichnis

`models` — Leerverzeichnis für trainierte KI-Modelle – **nicht löschen!**



DETAILS ZU WORKSHOPABLAUF UND IDEEN ZUR EINBETTUNG IN EIN CURRICULUM

WORKSHOP-ABLAUF

(Programmier- und weitere Erläuterungs/Theorieteile werden je nach Klassenstufe flexibel eingebaut in die einzelnen Schritte)

1. Kurze Einführung in Künstliche Intelligenz und Maschinelles Lernen

1. → Eingangsfolien kindgerecht aufbereiten → SuS-geeigneter Foliensatz zu diesem Lehrerfoliensatz

2. Durchlaufen eines Lern- und Inbetriebnahmezyklus für Maschinelles Lernen

1. Datensammelphase – Gamification!

1. Kinder teilen sich in mehrere Gruppen auf, ggf. auch jedes Kind mit eigenem Calliope
2. Spielen 10 Minuten lang Autorennen, generieren dabei Trainingsdaten
3. Trainingsdaten werden ggf. an einen Desktop übertragen (Pipeline-spezifisch) und zusammengeführt

2. Trainingsphase

1. Für jedes Set Trainingsdaten (einzeln, Gruppen- oder Klassen-Pool) wird ein ML-Modell trainiert

3. Testphase

1. ML-Modell wird mehrmals laufen gelassen, bis die KI einen Unfall baut. Statistischer Gesamt-Score
2. Trainingserfolg wird am Ende dargestellt; „Siegerehrung“

Um das Lernziel zu stützen:
Fokus auf Training & Test mit
unterschiedlichen Datenvolumina

3. Lernziele resümieren

1. Datenmenge und -qualität sind entscheidend für den Lernerfolg einer künstlichen Intelligenz
2. Option: Visualisieren des Anlernvorgangs eines künstlichen neuronalen Netzwerks, ggf. nur als Video:
<https://www.youtube.com/watch?v=Mr42DQHy3TI>



WORKSHOP-VARIANTEN

- Programmierinhalte, KI-Inhalte und Workshop-Dauer differenziert nach Klassenstufe, bspw.:
 - *Unterstufen:* Doppelstunde, keine eigene Programmierung, keine Erläuterung der Mathematik des ML-Algorithmus
 - *Mittelstufen:* halber Tag, wenig eigene Programmierung, grobe Erläuterung der Mathematik des ML-Algorithmus
 - *Oberstufen:* ganzer Tag, mehr eigene Programmierung, tiefere Erläuterung der Mathematik des ML-Algorithmus, alternative Algorithmen vorstellen (Orange Plattform)

Ideen hierzu auf der folgenden Folie

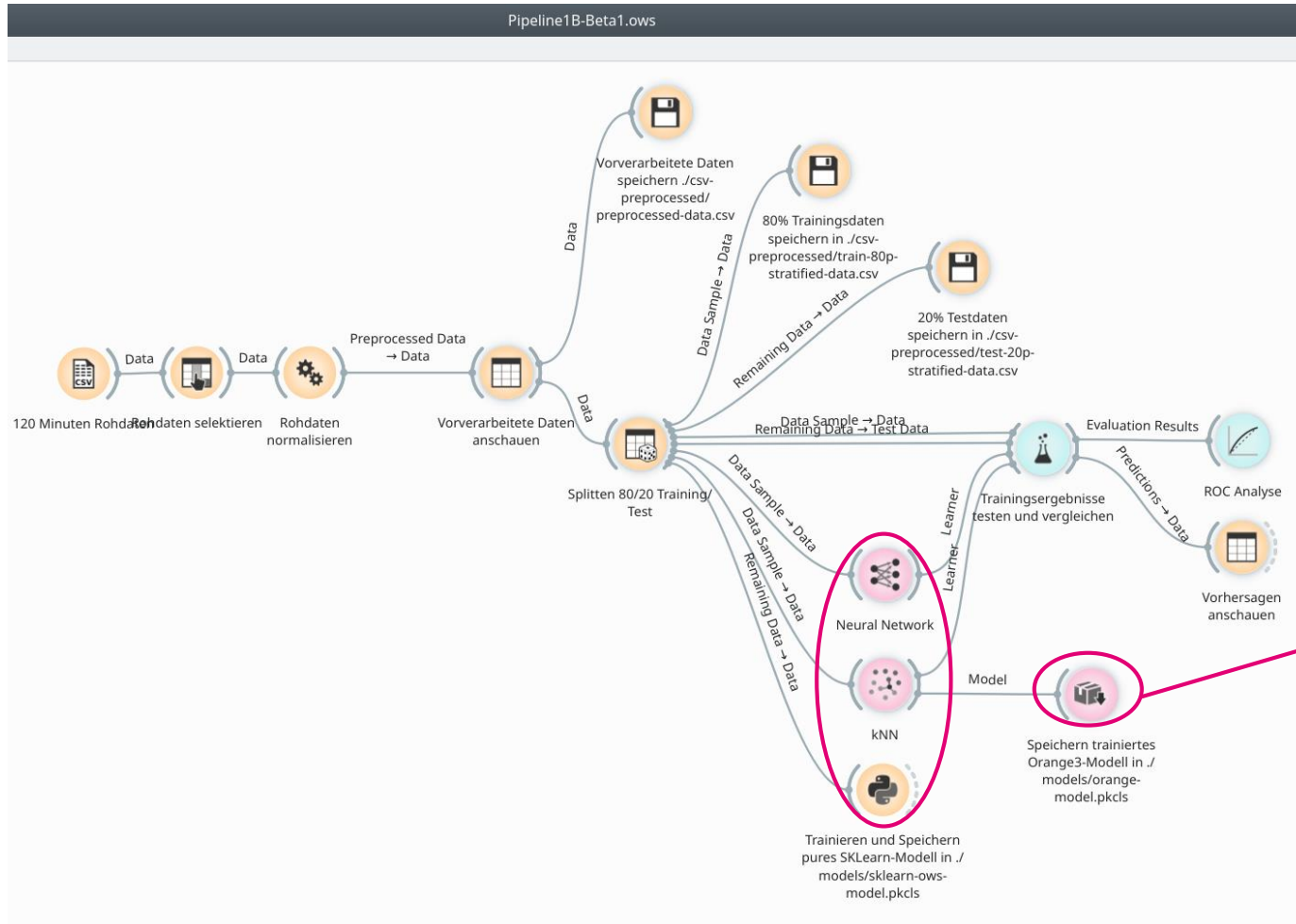
IDEEN ZUM EINBAU IN EIN INFORMATIK-CURRICULUM

MITTEL- UND OBERSTUFE

- Direkt in das komplexe KI-Thema einzusteigen stellt – je nach Wissensstand der SuS – eine Herausforderung dar.
- Daher ist es ggf. sinnvoll, eine Workshopserie um den KI-Teil herum zu bauen
- **Mögliche Curriculum**
 1. SuS programmieren zunächst selbst ein Autorennspiel wie das für dieses Projekt verwendete
Vorwissen wird aufgebaut: Wie funktioniert ein Autorennspiel auf dem Calliope (Codebasis Autorennspiel entwickeln)
 2. SuS programmieren zunächst selbst eine „klassische“ autonome Steuerung für das in Schritt 1 entwickelte Spiel
Vorwissen wird aufgebaut: Wie funktioniert eine „klassische“ Automatisierung (Codebasis für „klassische“ autonome Steuerung entwickeln)
 3. **Danach kann das KI@Calliope-Projekt starten mit dem aufgebauten Vorwissen, sodass das Lernziel auf das Thema „Künstliche Intelligenz“ fokussiert werden kann, ohne erst das aufgebaute Vorwissen vermitteln zu müssen.**
- **Ergänzungen für Oberstufe**
 - Vertiefung des Themas, z.B.
 - Statt neuronale Netze andere ML-Modelle ausprobieren (RandomForest, SVM, NaiveBayes, ...)
 - Komplexere neuronale Netze ausprobieren mit MIT DeepTraffic



BEISPIEL OBERSTUFE: ORANGE PLATFORM



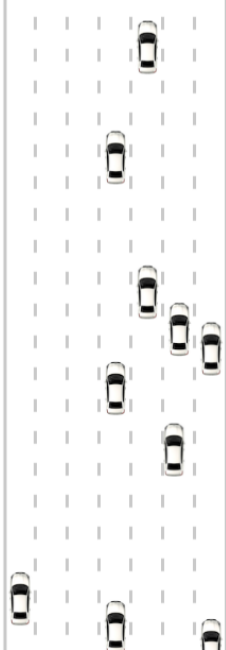
- Orange ist eine für das studentische Umfeld entwickelte grafische Plattform für Datenanalyse und maschinelles Lernen
- Unser Projekt integriert auch mit einem für unser Projekt entwickelten Orange Workflow (.ows Datei)
 - Eingabedaten in den Workflow sind die mit dem Datensammler gesammelten Logdaten der Calliope Minis
 - Ausgabe des Workflows ist ein **beliebiges** KI-Modell, mit dem das autonome Calliope Mini Auto auch gesteuert werden kann.
- Dies eröffnet die Option, für die Lösung des Problems nicht nur ein neuronales Netzwerk, sondern auch andere Algorithmen des maschinellen Lernens zu verproben und in die mathematischen/informatischen Hintergründe einzusteigen.

BEISPIEL OBERSTUFE: VERTIEFUNG FÜR INTERESSIERTE

- <https://selfdrivingcars.mit.edu/deeptraffic/>

MIT 6.S094: Deep Learning for Self-Driving Cars

[Home](#) **DeepTraffic** [MIT Deep Learning](#) [Resources](#) [View Profile](#) [Register](#) [Login](#)



Speed:
80 mph

DeepTraffic

[Visualization](#) - [Leaderboard](#) - [Documentation](#) - [Paper](#) - [GitHub](#)

Americans spend 8 billion hours stuck in traffic every year.
Deep neural networks can help!

```
1  
2 <![CDATA[  
3  
4 // a few things don't have var in front of them - they update already  
   existing variables the game needs  
5 lanesSide = 0;  
6 patchesAhead = 1;  
7 patchesBehind = 0;  
8 trainIterations = 10000;  
9
```

[Apply Code and Reset Net](#) [Save Code and Net to File](#) [Load Code/Net from File](#)
[Submit Model to Competition](#)