

WISSEN ZU KÜNSTLICHER INTELLIGENZ SPIELERISCH IN DIE SCHULEN BRINGEN MIT DEM CALLIOPE MINI

FERENC HECHLER, MIRKO JELINEK, CHRISTIAN SCHILLER, DIRK WOLTERS

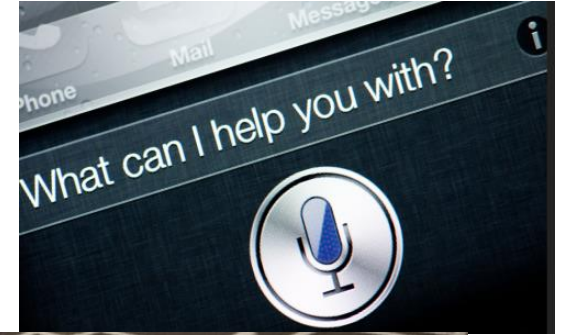


ERLEBEN, WAS VERBINDET.

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

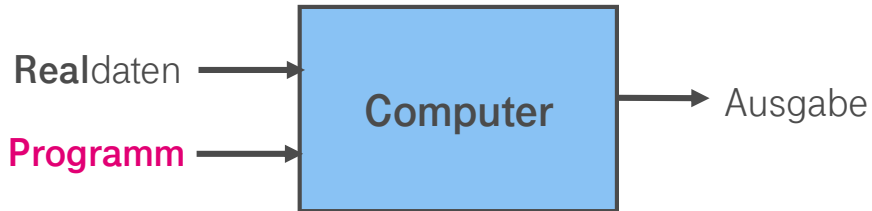
- **KI-Workshop für Kinder**

- 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

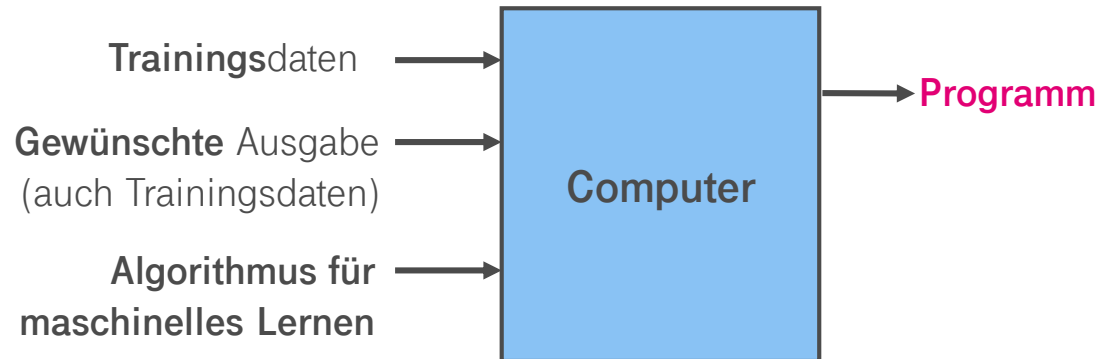


„Software 1.0“

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

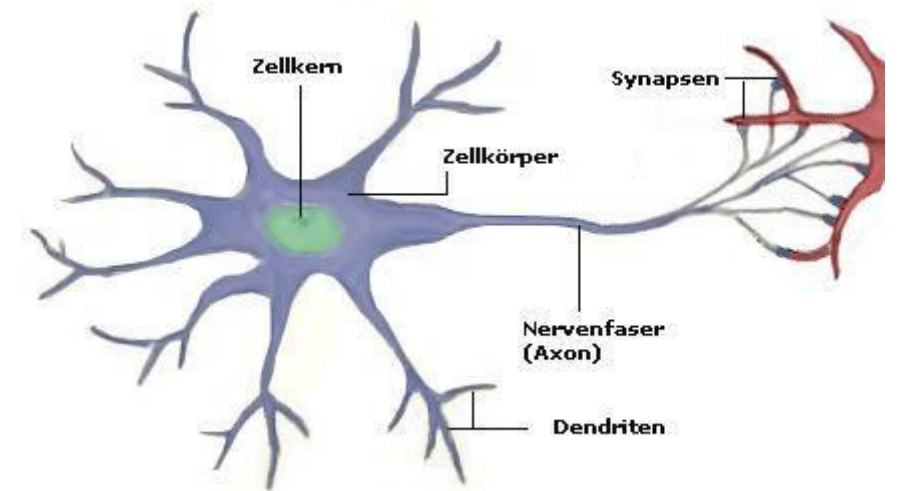
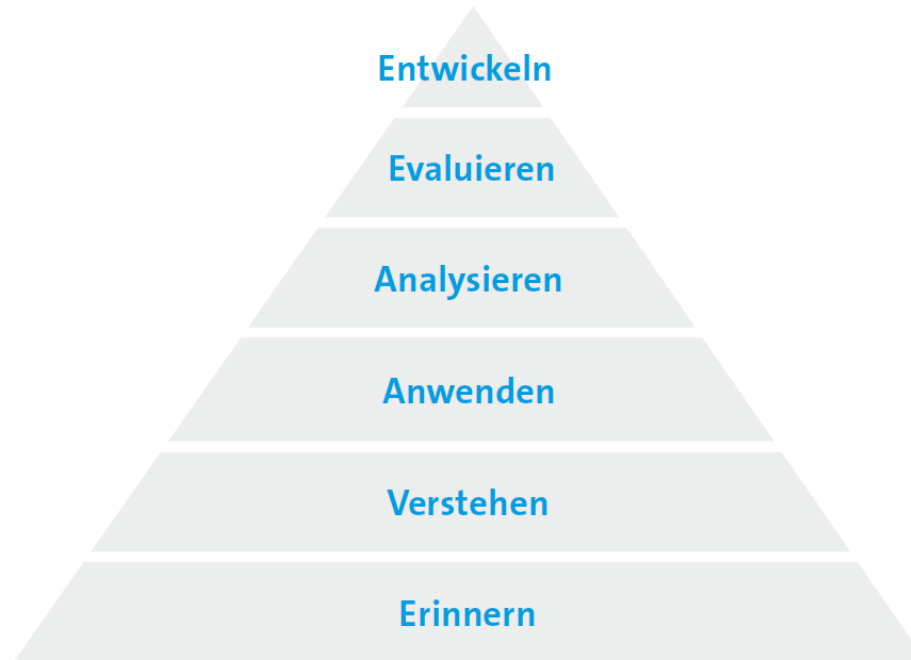
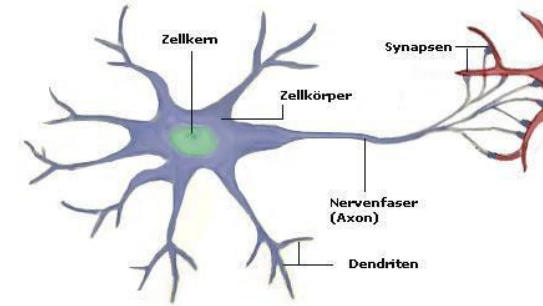


Abbildung 2: Hierarchische Klassifikation kognitiver Fähigkeiten²⁸

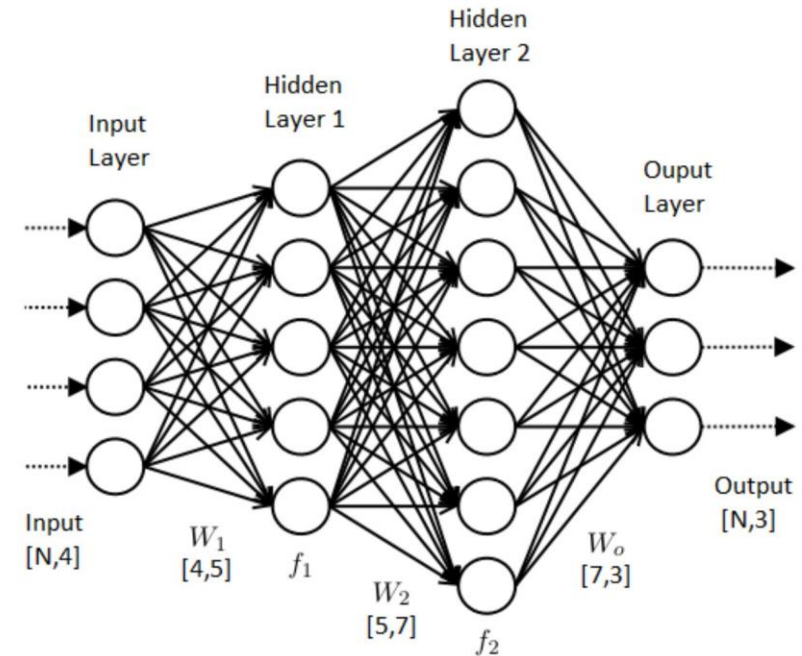
KÜNSTLICHE „INTELLIGENZ“

- 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



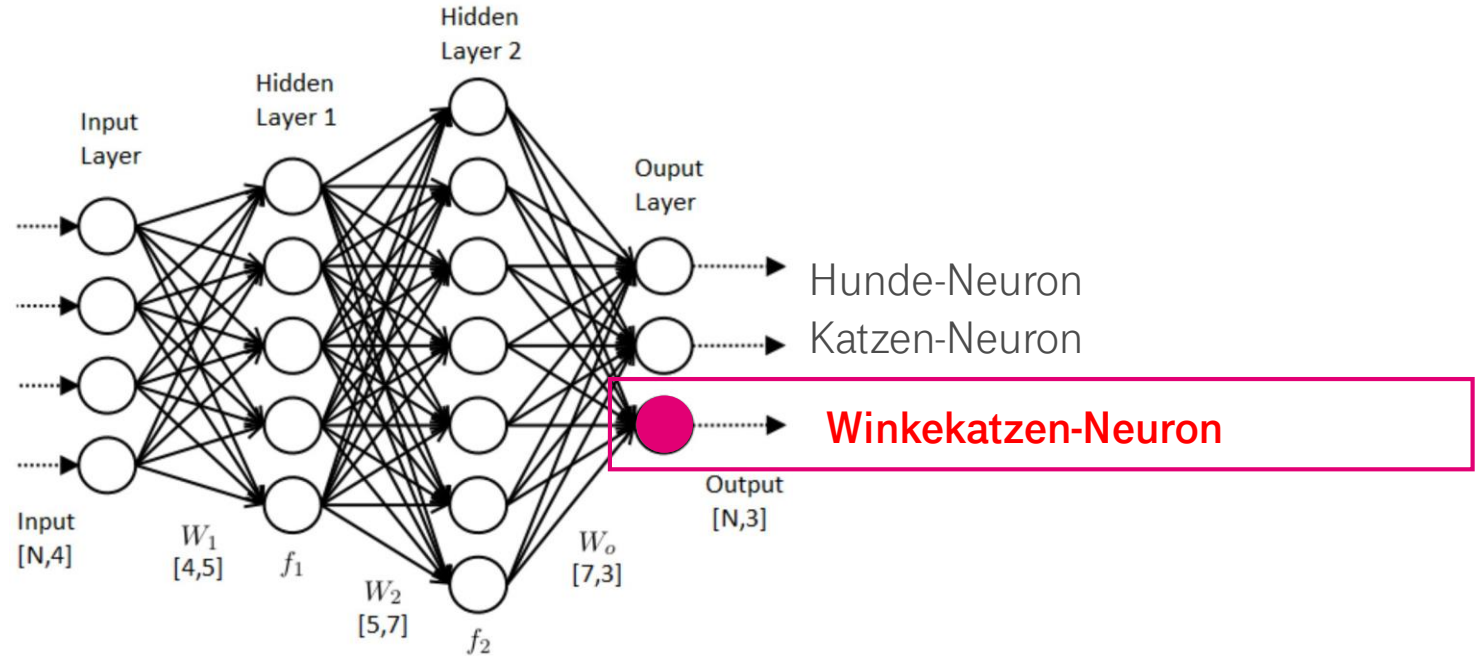
Künstliche Neuronen



KNN – GRUNDSÄTZLICHE FUNKTIONSWEISE



Bildquelle

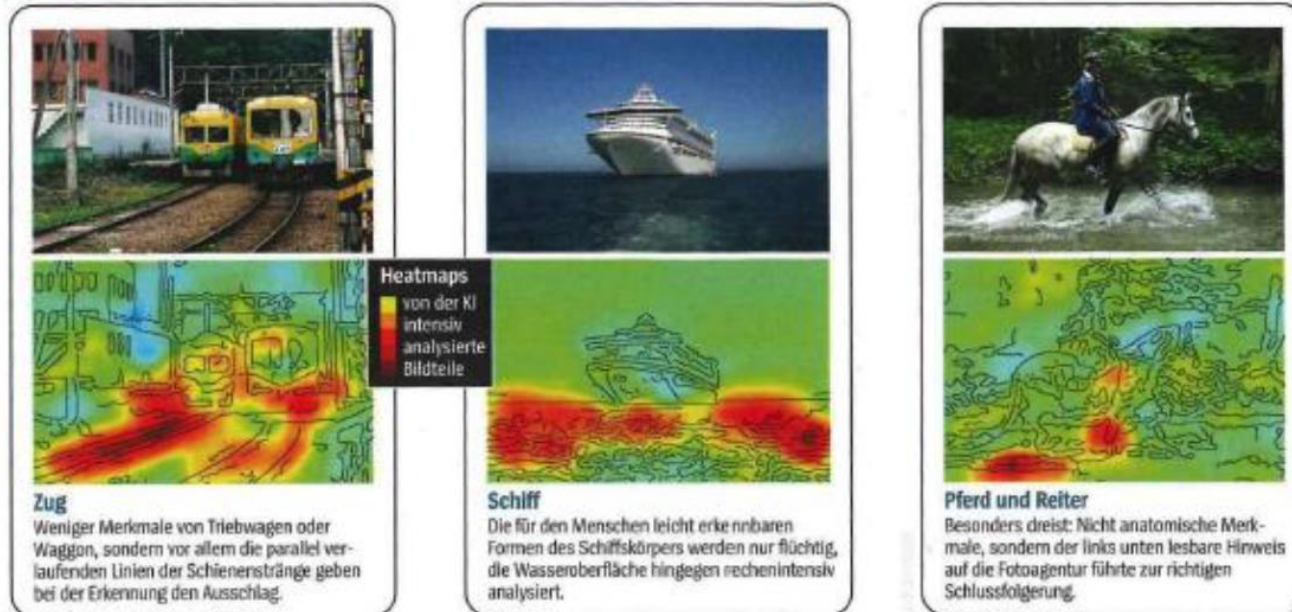


Bei einem angelernten/trainierten KNN sind die Gewichte der Neuronenverbindungen (W_x) so gesetzt (angelernt/trainiert) worden, dass bei einem eingegebenen Winkekatzen-Bild immer das „Winkekatzen-Neuron“ aktiviert wird

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!*)

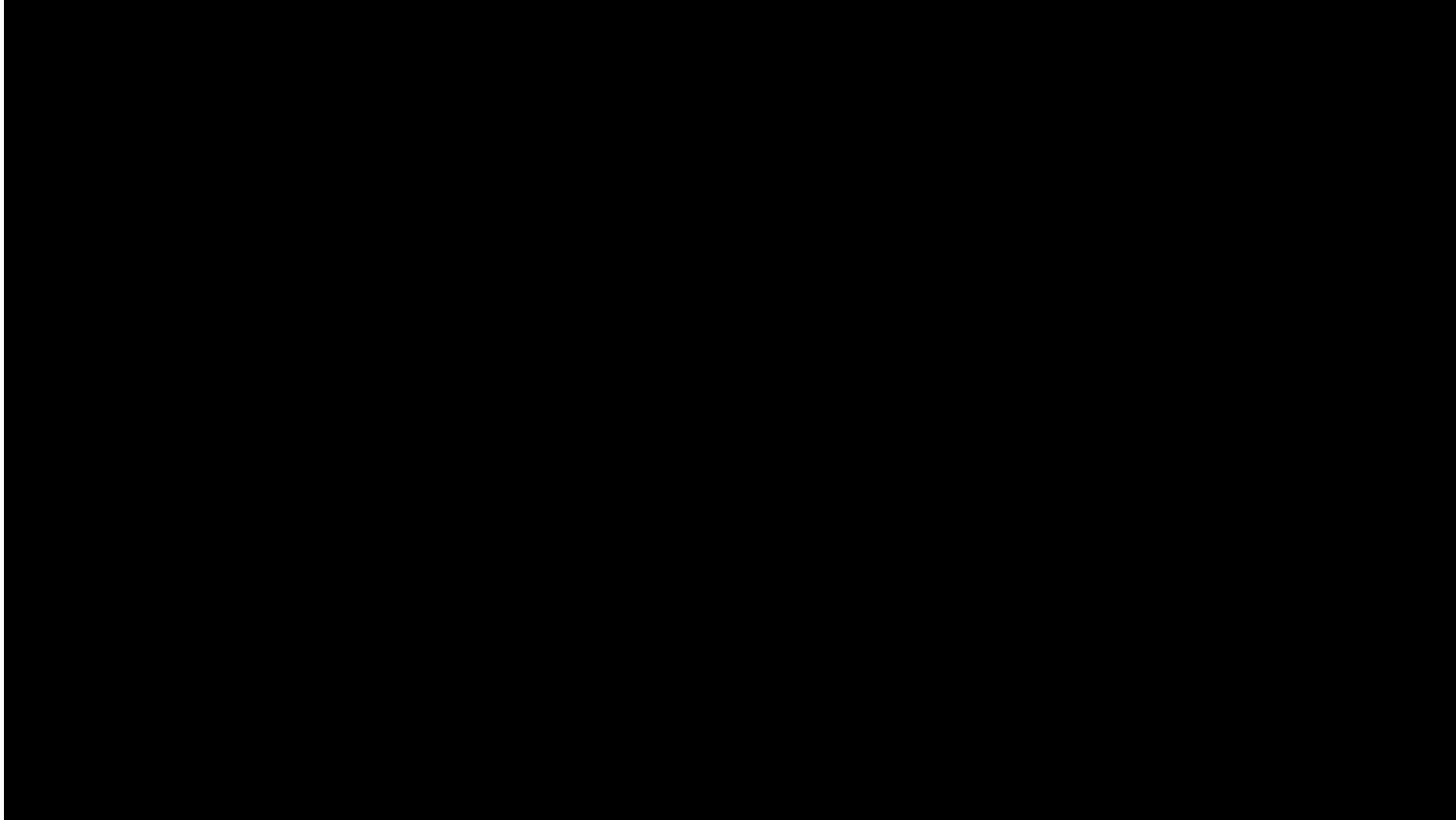
Züge → Gleise

Schiffe → Wasserflächen

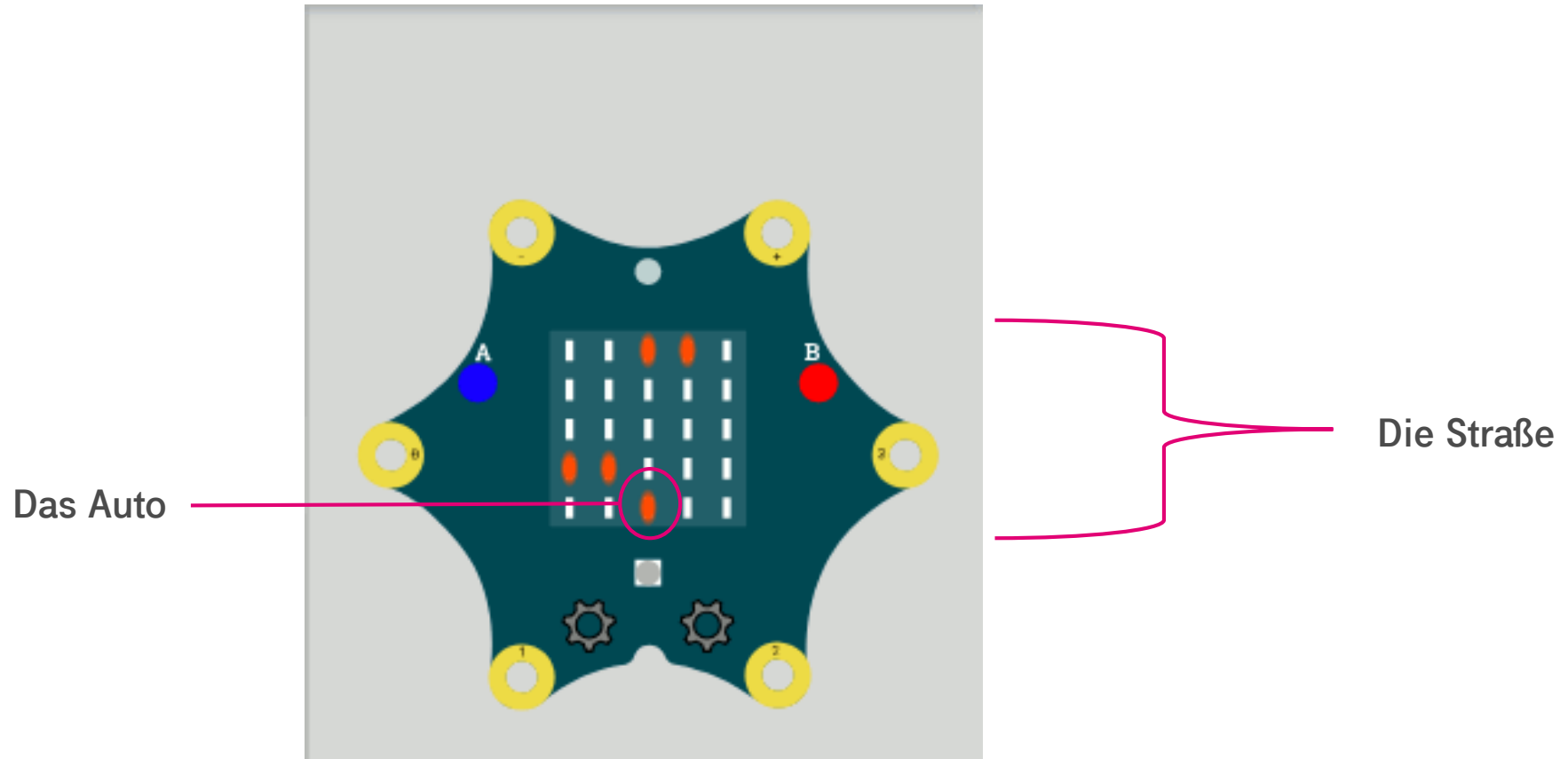
Pferde+Reiter → Logo der Fotoagentur, die die Bilder bereit gestellt hat

KI FÜR AUTONOMES FAHREN

So “sieht” ein Tesla Autopilot die Straßen von Paris - https://www.youtube.com/watch?v=_1MHGUC_BzQ



(AUTONOMES) AUTO AUF DEM CALLIOPE MINI

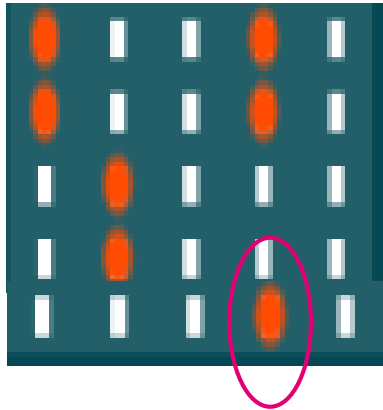


VERGLEICH MIT ECHTEM AUTONOMEN FAHREN

(„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 Autokonzern machen prinzipiell nichts anderes, müssen dabei „nur“ eine „etwas“ höhere Komplexität bewältigen (siehe Foto rechts)

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

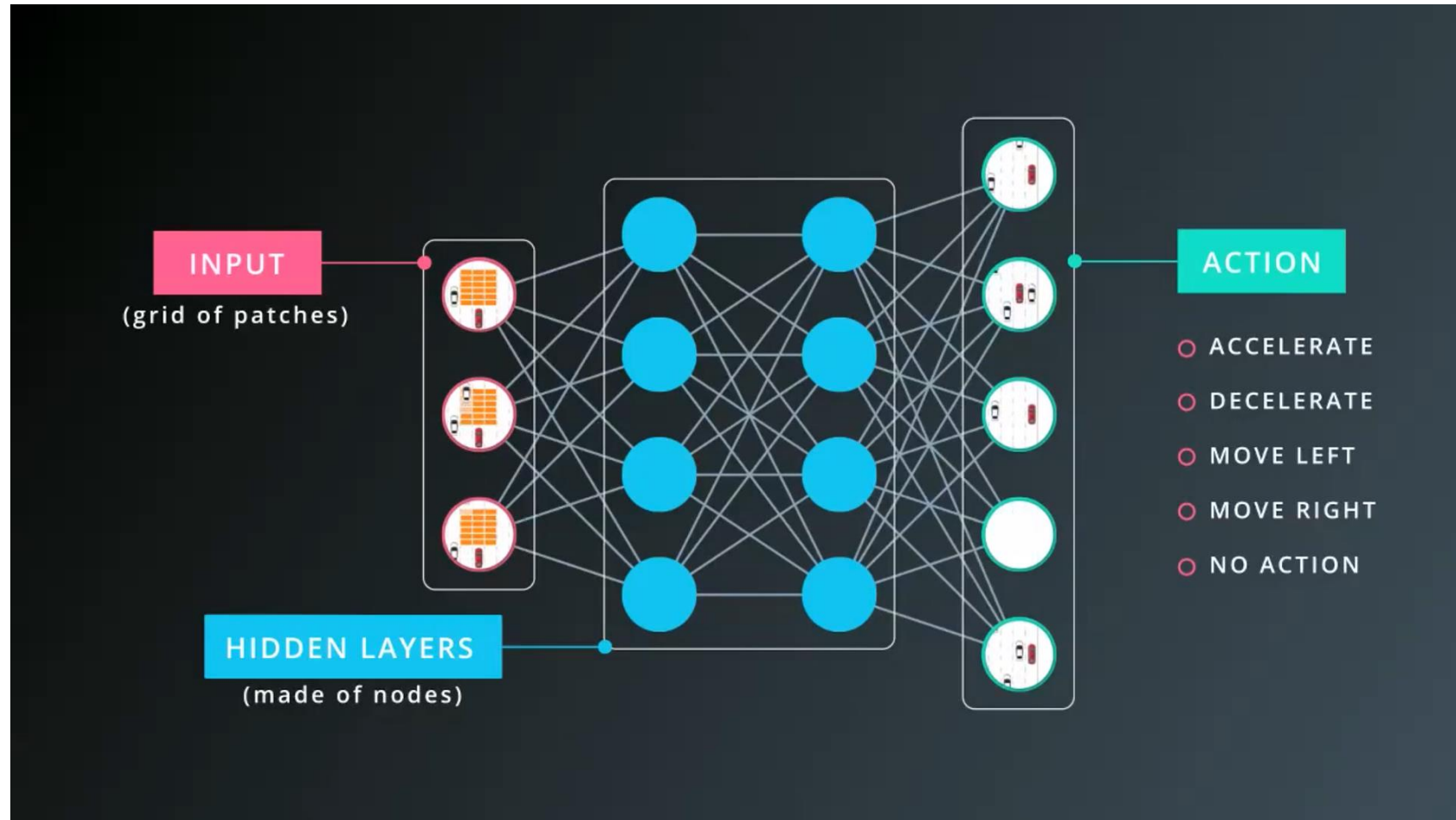


Ein echtes Auto
sieht sowas vor sich:



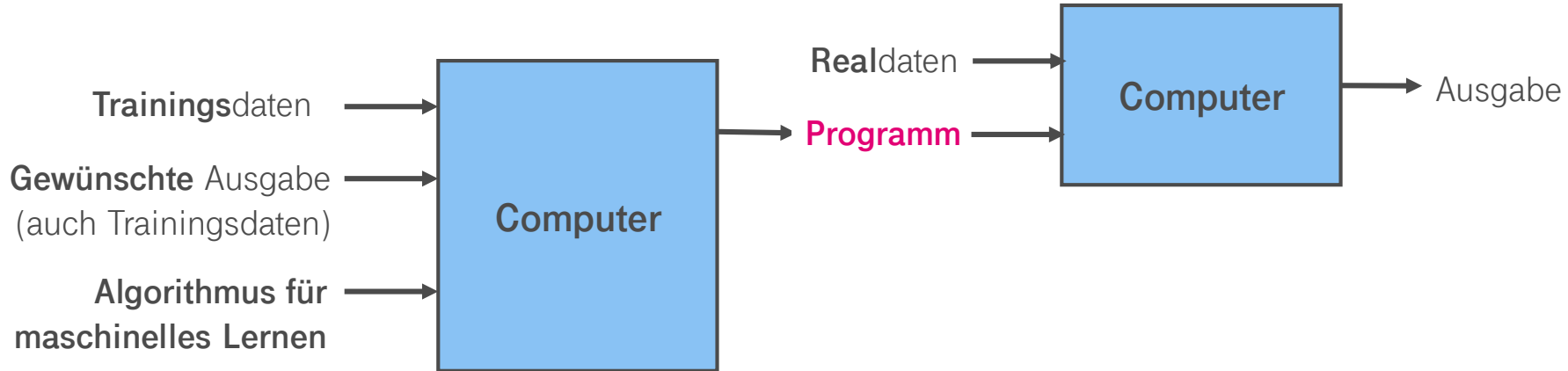
Dieser Pixel hier ist das
steuerbare „Calliope-Auto“

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

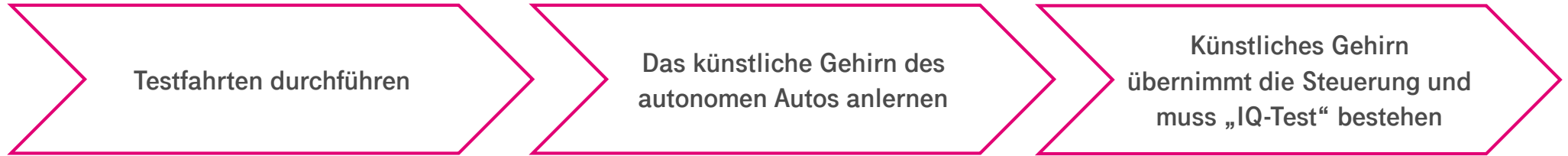
Trainingsdaten sammeln

ML Modell trainieren

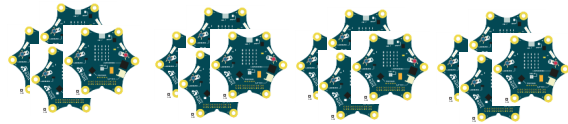
ML Modell anwenden und testen



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



Variante 1A & 1B

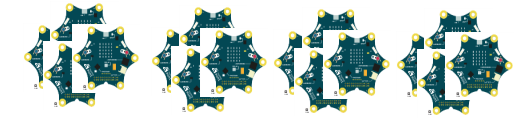


Pooling der Trainingsdaten von Schülergruppen



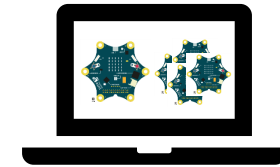
mit einem „großen“ künstlichen neuronalen Netzwerk auf einem Desktop-Rechner

1A



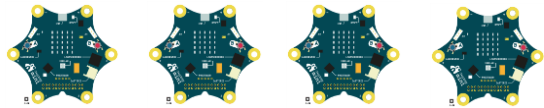
Messung des Trainingserfolgs pro Schülergruppe auf Calliope Mini

1B

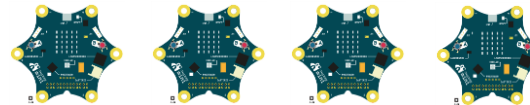


Messung des Trainingserfolgs „im Backend“

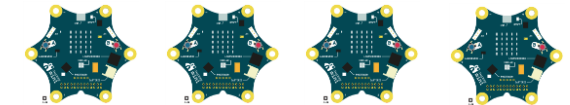
Variante 2



Jeder Schüler individuell



mit einem „kleinen“ künstlichen neuronalen Netzwerk auf dem Calliope



Messung des Trainingserfolgs pro Schüler auf Calliope Mini

REALISIERUNGSSTATUS STAND 12/2019

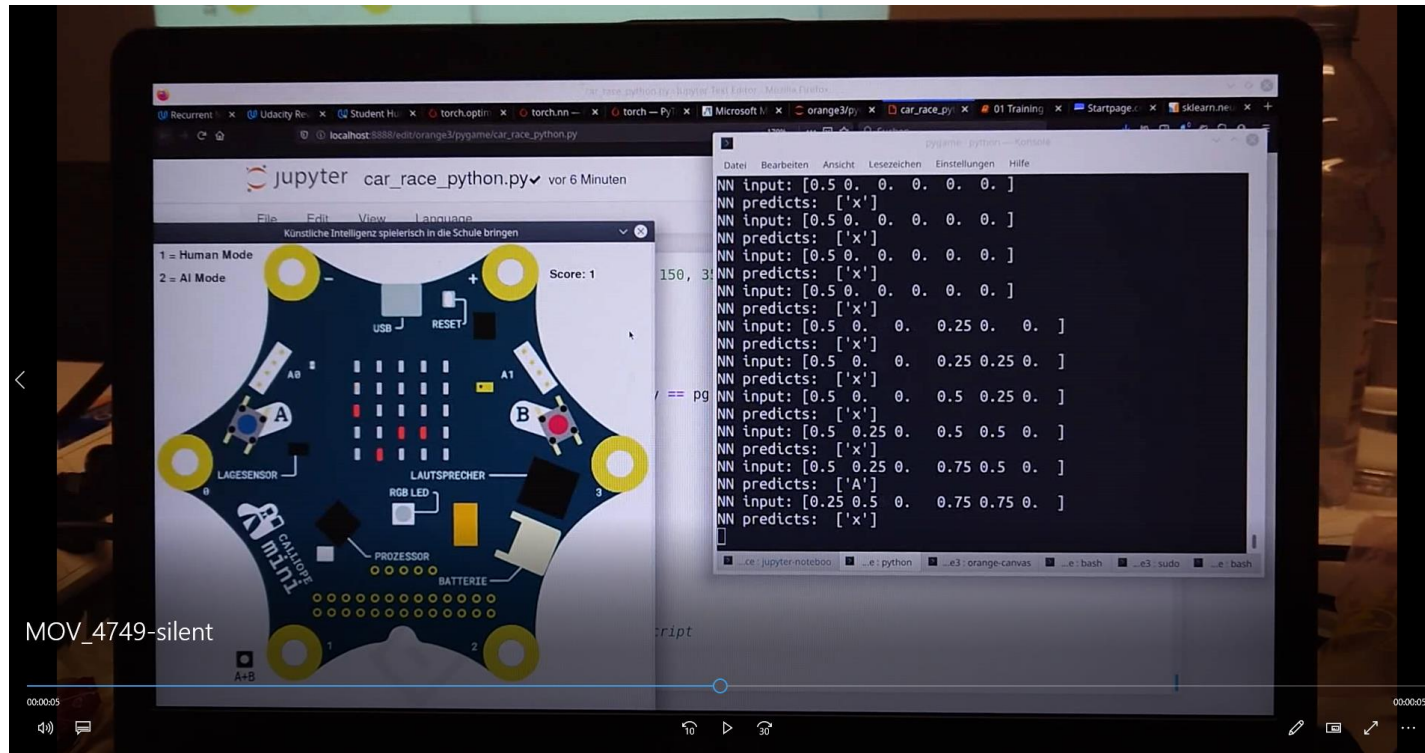
Variante	Hardware	Software	Trainingsdaten sammeln	ML Modell trainieren	ML Modell anwenden und testen	Status
1A	Calliope, Desktop					Noch nicht begonnen
1B	Calliope, Desktop	Calliope Makecode, PuTTY, Jupyter Notebook, [Orange3], Scikit-Learn	Calliope, Desktop	Desktop	Desktop	Erster komplett funktionsfähiger Prototyp erstellt!
2	Calliope	Calliope Makecode	Calliope	Calliope	Calliope	in Arbeit



AKTUELLER PROTOTYP (12/2019)

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



ERLEBEN, WAS VERBINDET.

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

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

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

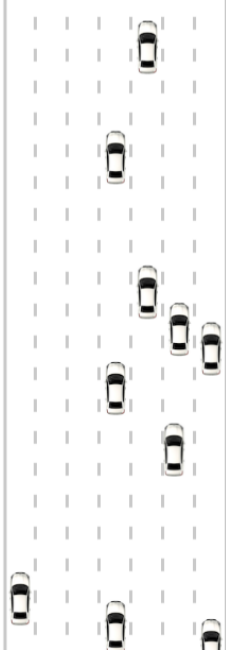


BEISPIEL OBERSTUFE: VERTIEFUNG FÜR INTERESSIERTE

- <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](#)

DISCLAIMER & CONTACT

- **Contact the Author**

- No redistribution of slides or slide deck without the author's consent. Don't hesitate to contact!
- Christian Schiller, Senior IT & Enterprise Architect, Deutsche Telekom IT GmbH
 - christian.schiller@t-systems.com / +49-151-1822-4938 /  [@LinkedIn](#) /  [@Xing](#) /  [@Twitter](#)
 - BASIC 2.0 and 6502 Assembly since 1989
 - Open Group Master Certified IT Architect since 2009
 - TOGAF 9 Certified since 2014
 - Big Data Architecture since 2015
 - AI/ML Architecture since 2016
 - Udacity Nanodegrees:
 - Data Analyst (2018), Machine Learning Engineer (2019)
 - Deep [Reinf.] Learning (in progress)
 - Current AI/ML technical skills:



TensorFlow



PyTorch



ERLEBEN, WAS VERBINDET.

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