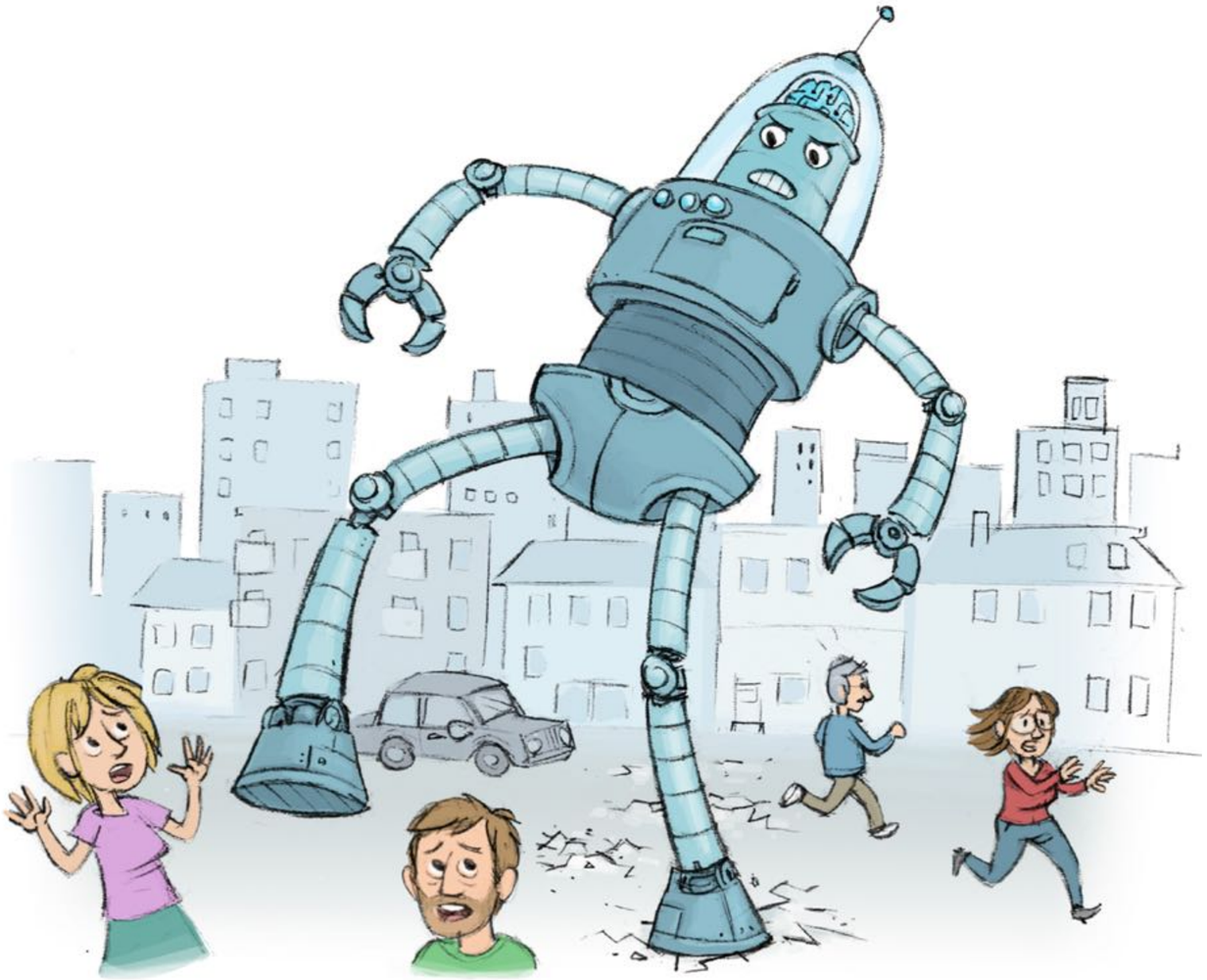
Warum meine KI die Menschheit auslöscht

Christopher Keibel

- Software Engineer @ Karakun
- Master Student Data Science
- Machine Learning Enthusiast

Inhalt

- KI für dummies
- Betrunkene KIs
- PacMan (leider auch betrunken)
- Stop-Button Problem



KI für dummies

Künstliche Intelligenz

- „Schlaue“ Computer Programme

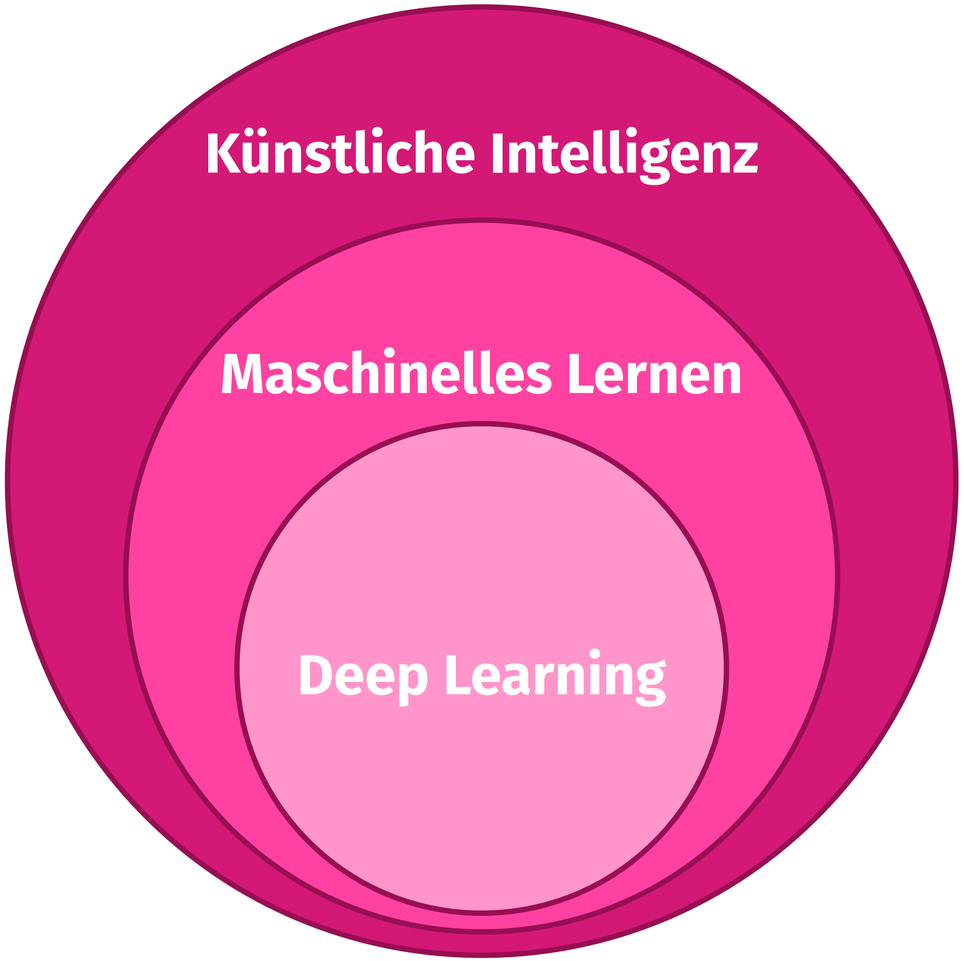
Künstliche Intelligenz

- „Schlaue“ Computer Programme

Künstliche Intelligenz

- „Schlaue“ Computer Programme
  - Selbstständig lernende Programme

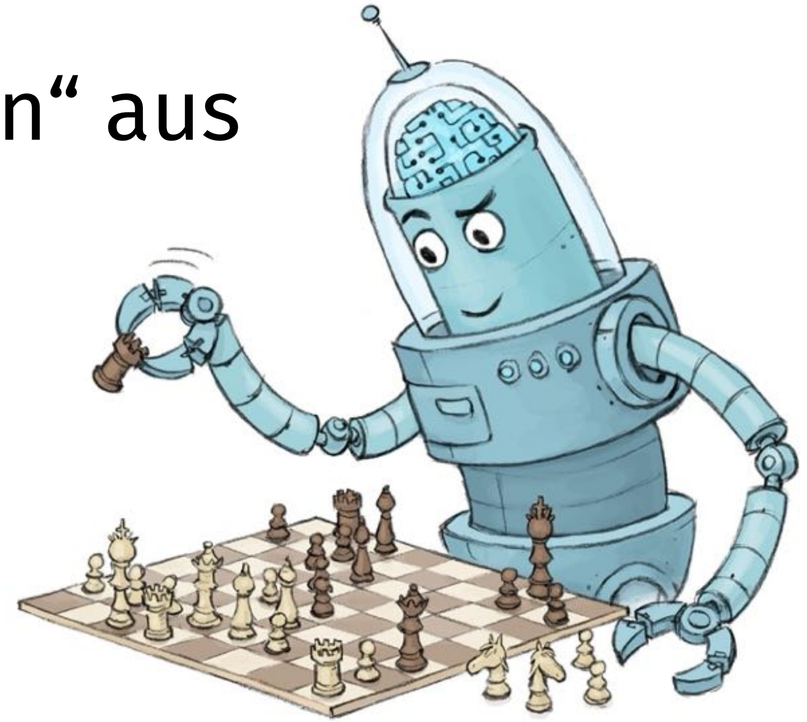
Künstliche Intelligenz



- „Schlaue“ Computer Programme
  - Selbstständig lernende Programme
- Künstliche Neuronale Netze

Maschinelles Lernen

- Algorithmen „lernen“ sich selbstständig in einer Aufgabe zu verbessern
- Verbesserungen erfolgen durch „Lernen“ aus Datensätzen



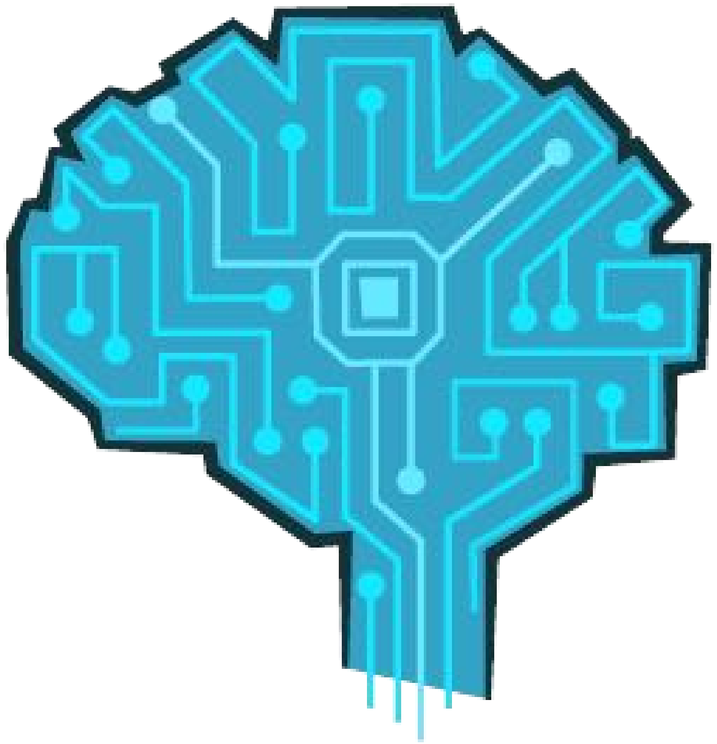
Der Datensatz

- Datensatz besteht aus Eingabedaten x und zugehörigen Ausgaben y*

*im überwachten Lernen (unbewachtes Lernen kein y -> Daten werden geclustert um Strukturen zu erkennen)

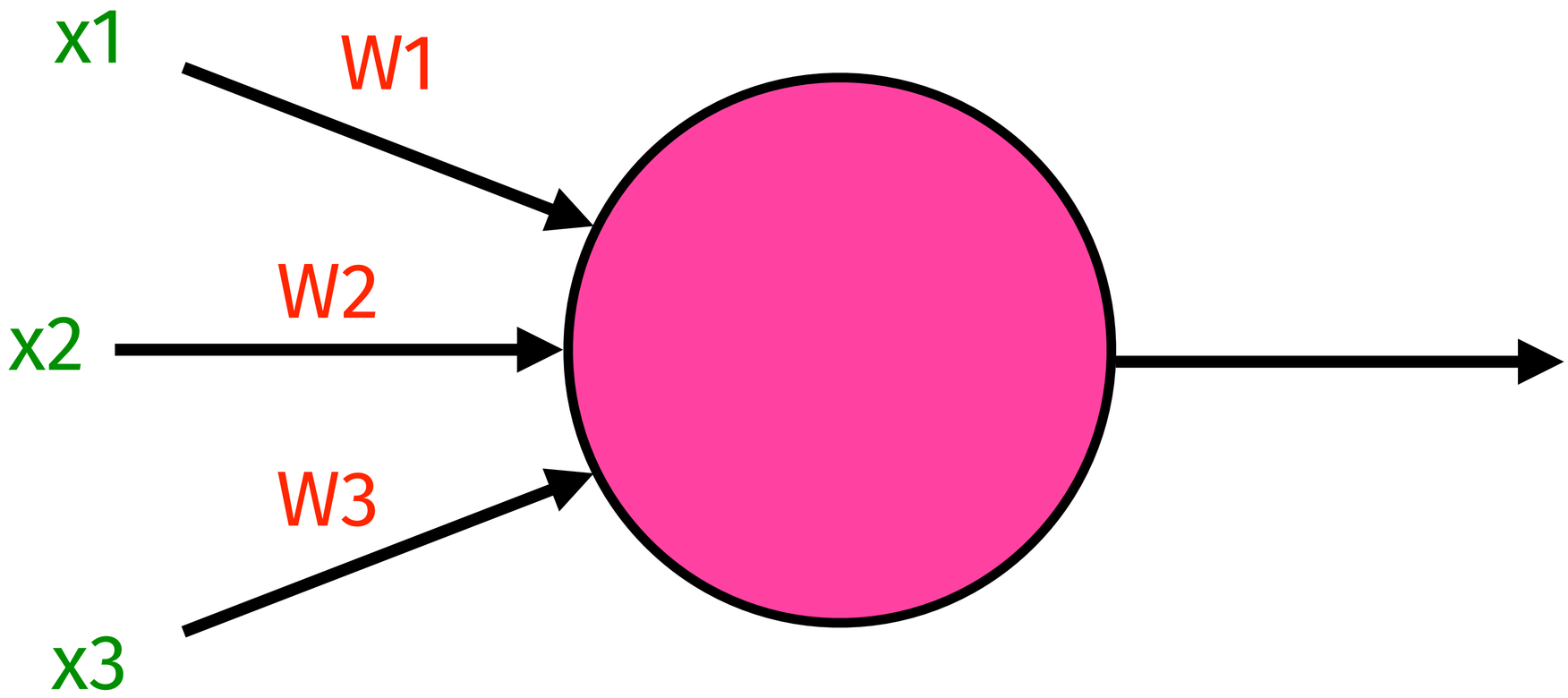
Deep Learning

- Lösungsfindung durch künstliche **neuronale Netze**
- Künstliche neuronale Netze bestehen aus einer Vielzahl kleiner Recheneinheiten (künstliche Neuronen)

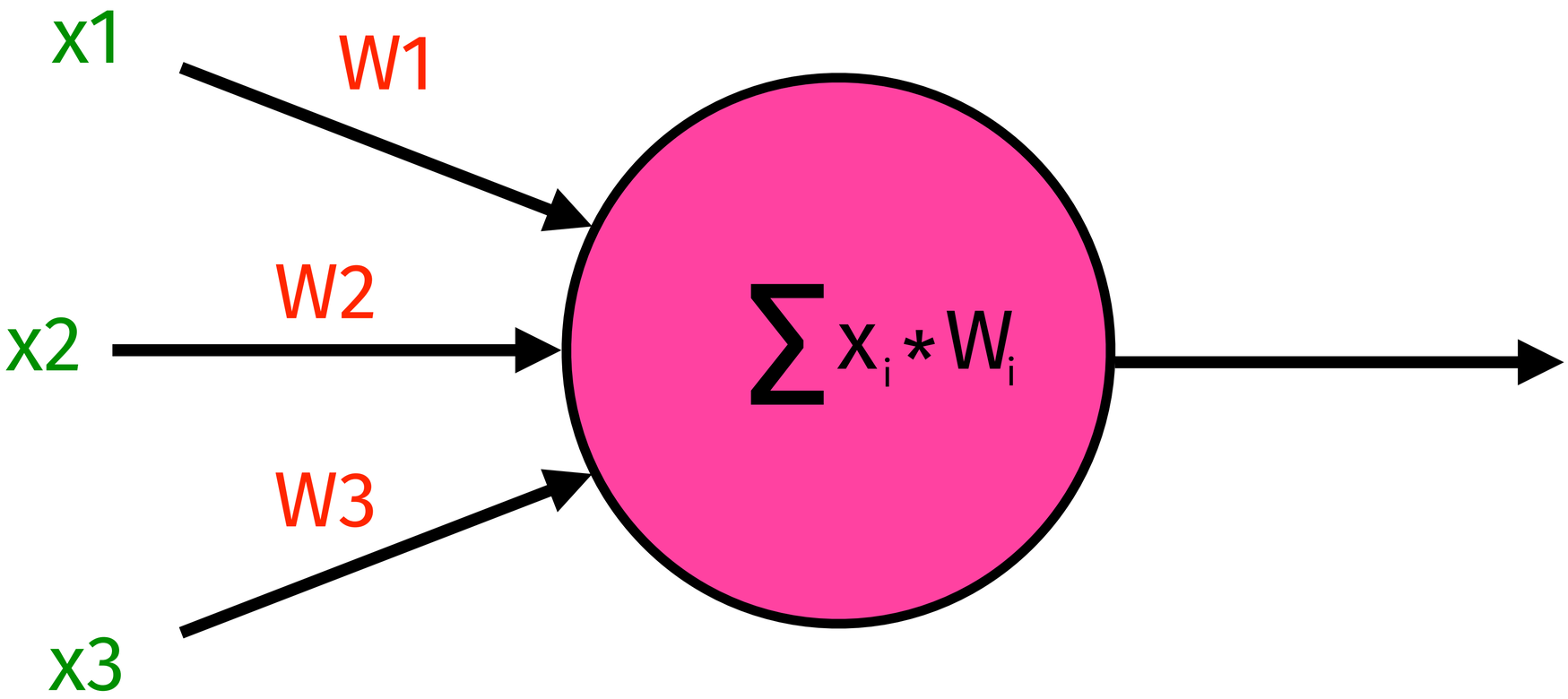


Künstliches Neuron

x1 * W1
x2 * W2
x3 * W3

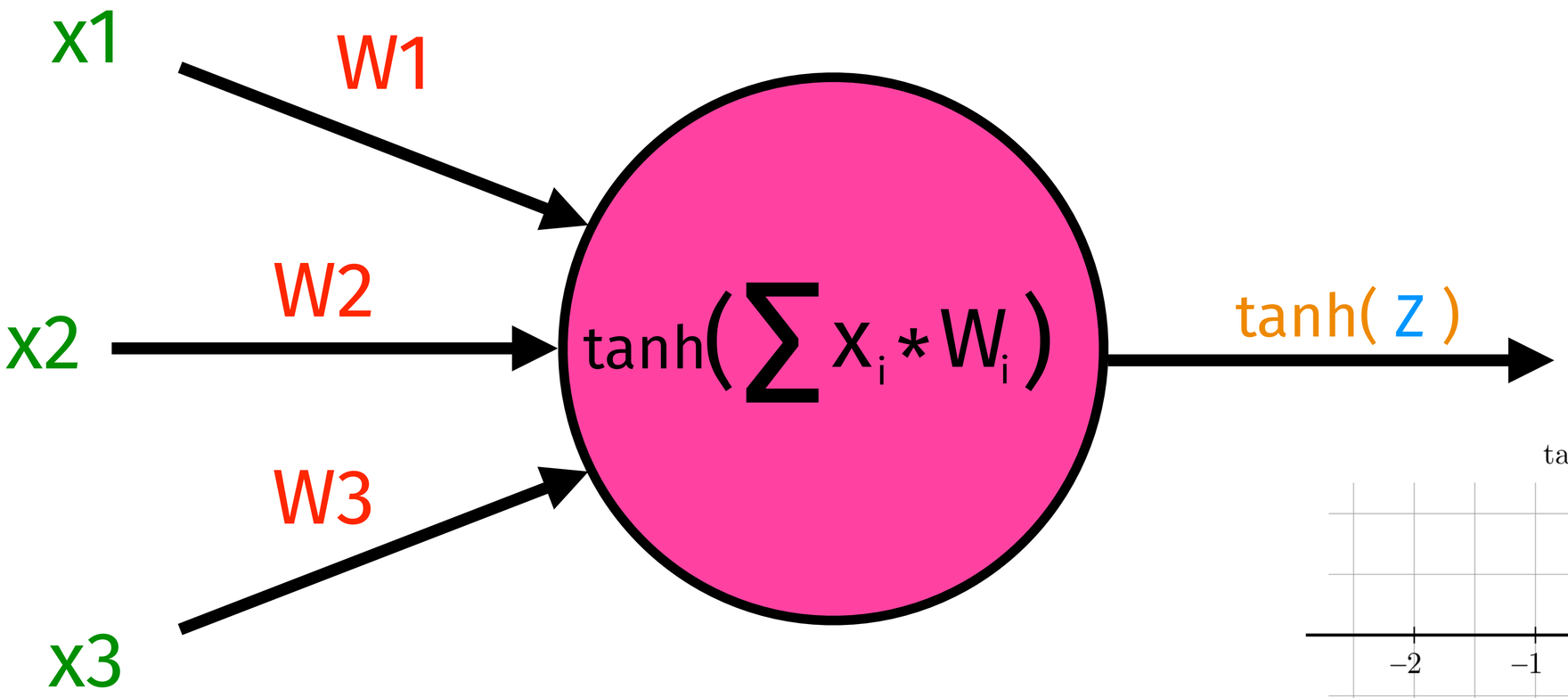


Künstliches Neuron

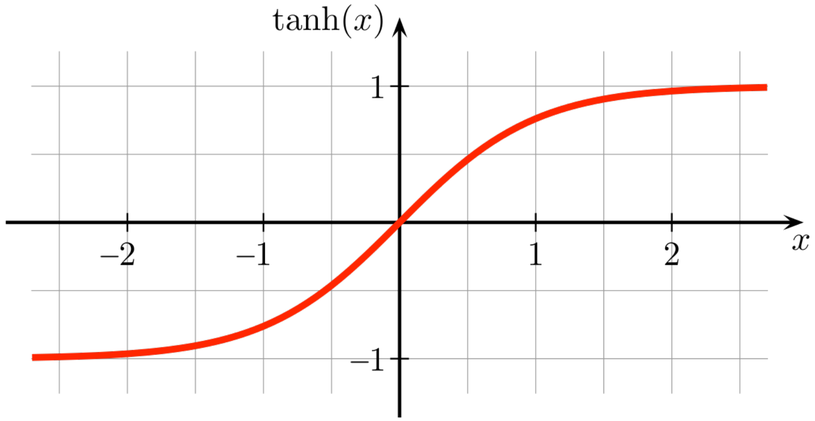


$$x1 * W1 + x2 * W2 + x3 * W3 = Z$$

Künstliches Neuron

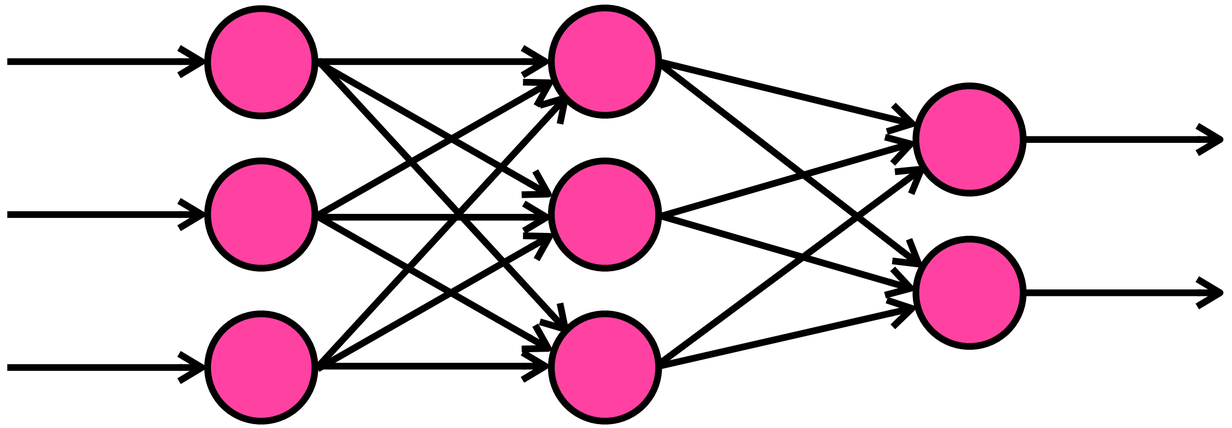


x1 * W1
+ x2 * W2
+ x3 * W3
= tanh(Z)



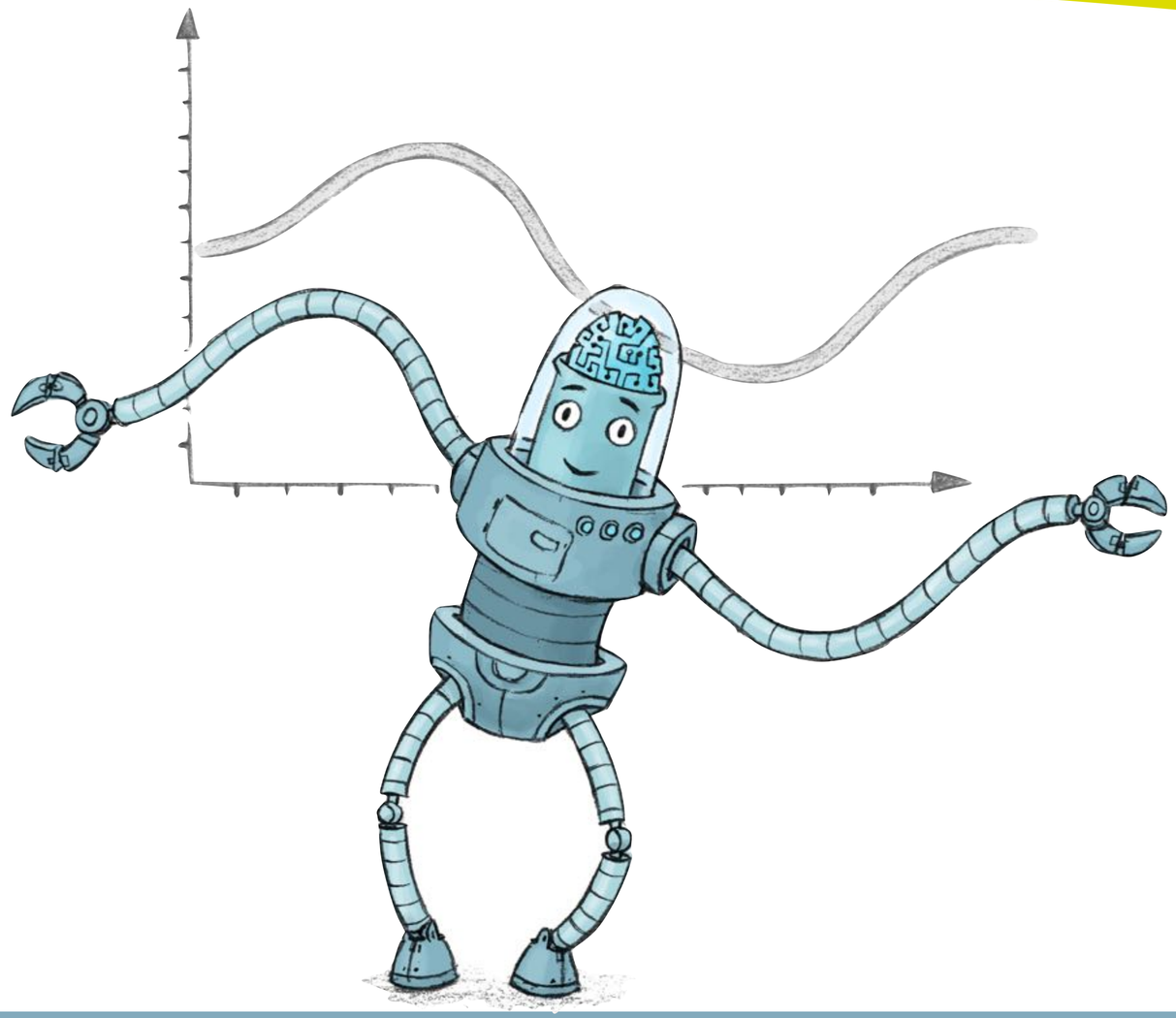
Deep Learning

- Recheneinheiten (Neuronen) definieren einfache mathematische Operationen
- Zusammen können solche Recheneinheiten eine komplexe Funktion abbilden

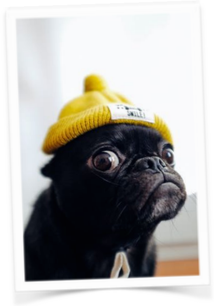


Was wird gelernt?

- Das Netz lernt eine Funktion zu „imitieren“
- Eingaben x wird auf Ausgaben y abgebildet

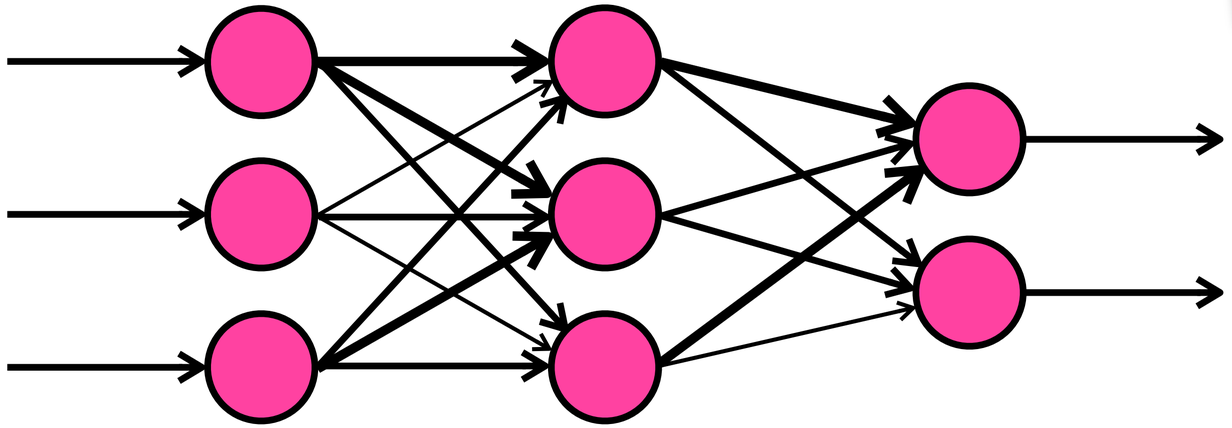


Wie wird gelernt?



KATZE

HUND

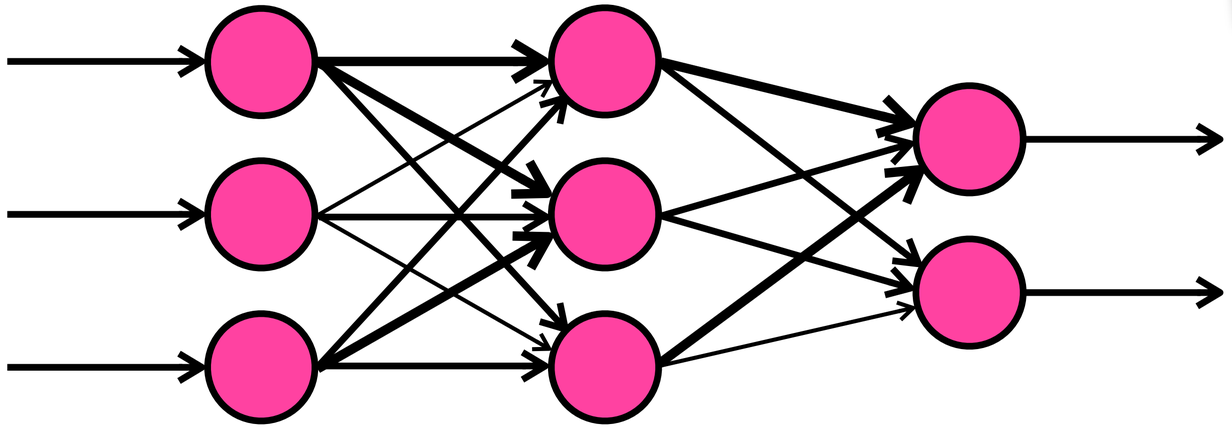


Wie wird gelernt?

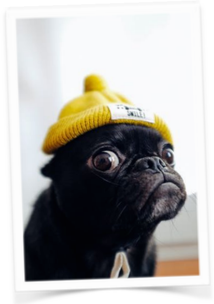
$$= \begin{bmatrix} 1 & 2 & 5 \\ 0 & 2 & 2 \\ 2 & 2 & 4 \end{bmatrix}$$

KATZE $= \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

HUND $= \begin{bmatrix} 0 \\ 1 \end{bmatrix}$



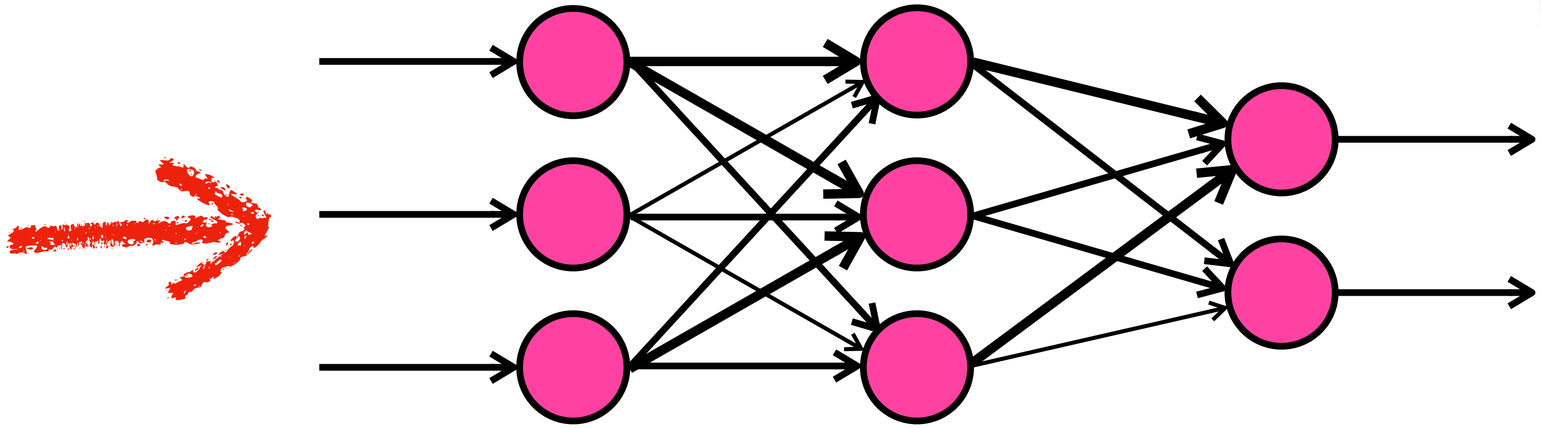
Wie wird gelernt?



KATZE $= \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

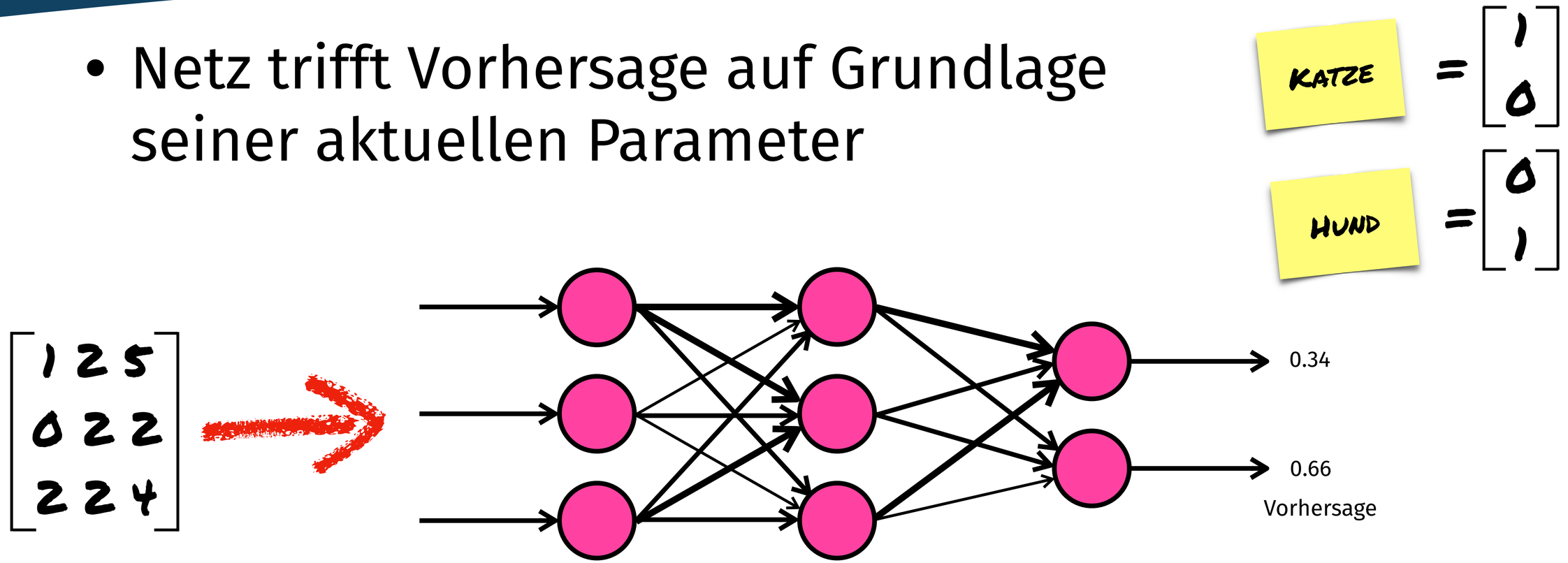
HUND $= \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$\begin{bmatrix} 1 & 2 & 5 \\ 0 & 2 & 2 \\ 2 & 2 & 4 \end{bmatrix}$



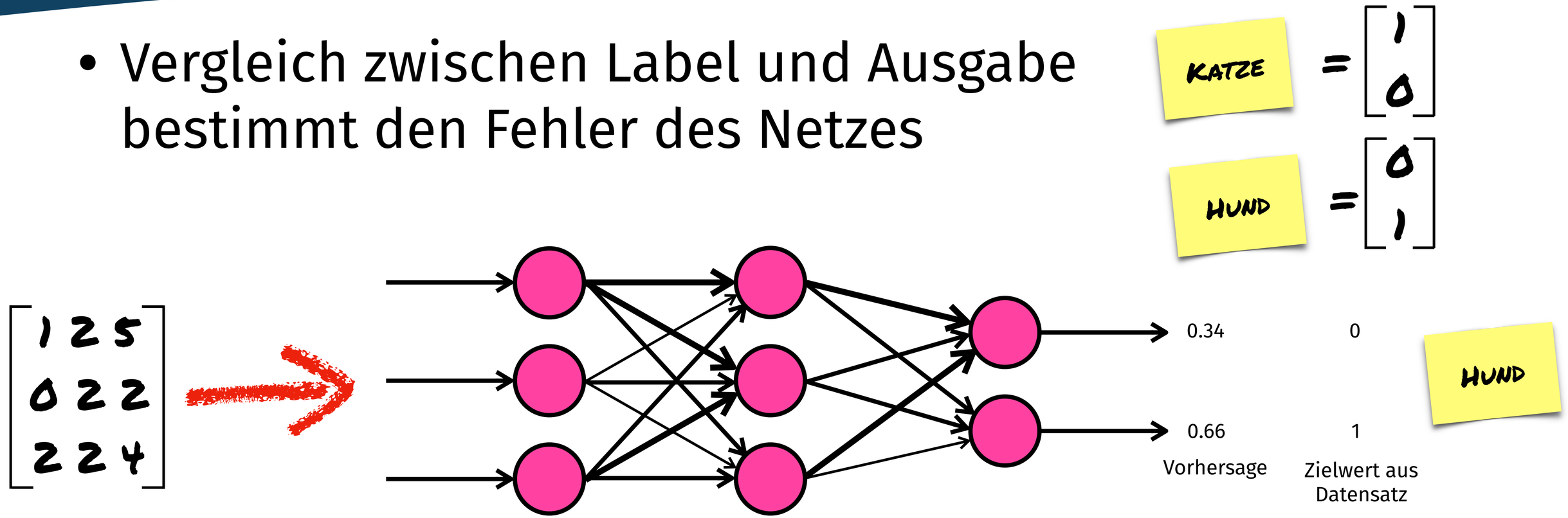
Wie wird gelernt?

- Netz trifft Vorhersage auf Grundlage seiner aktuellen Parameter



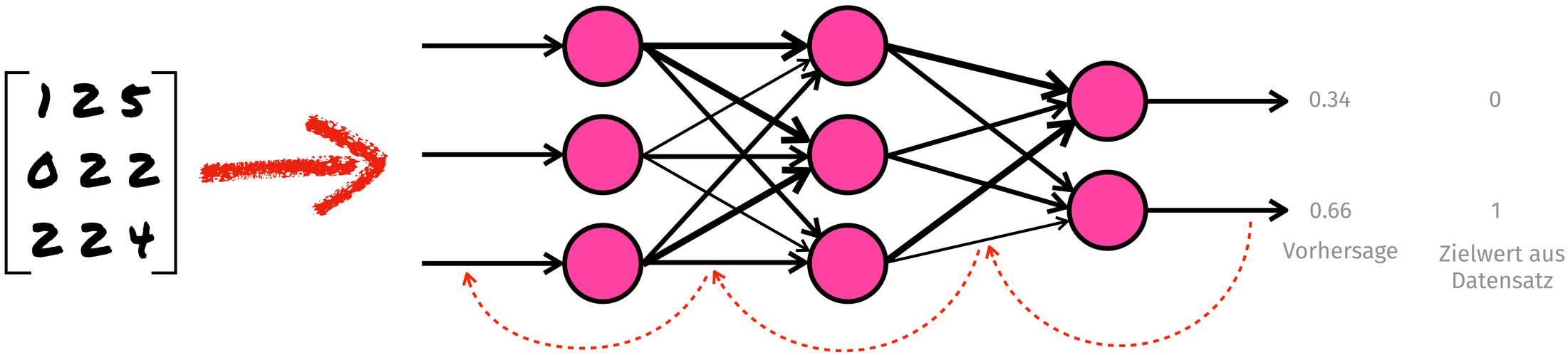
Wie wird gelernt?

- Vergleich zwischen Label und Ausgabe bestimmt den Fehler des Netzes

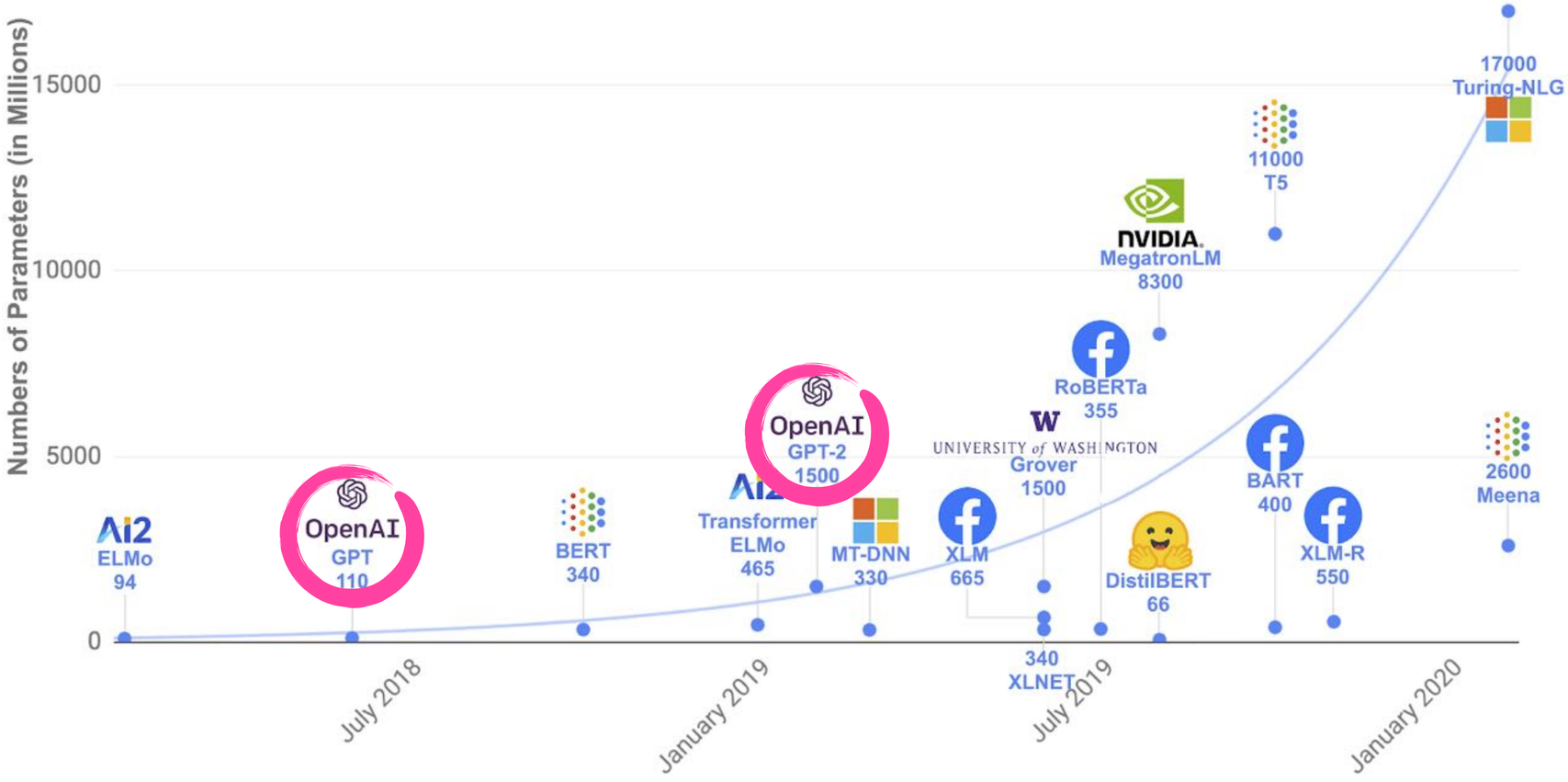


Wie wird gelernt?

- Verbindungsgewichte anpassen um Fehler zu verringern

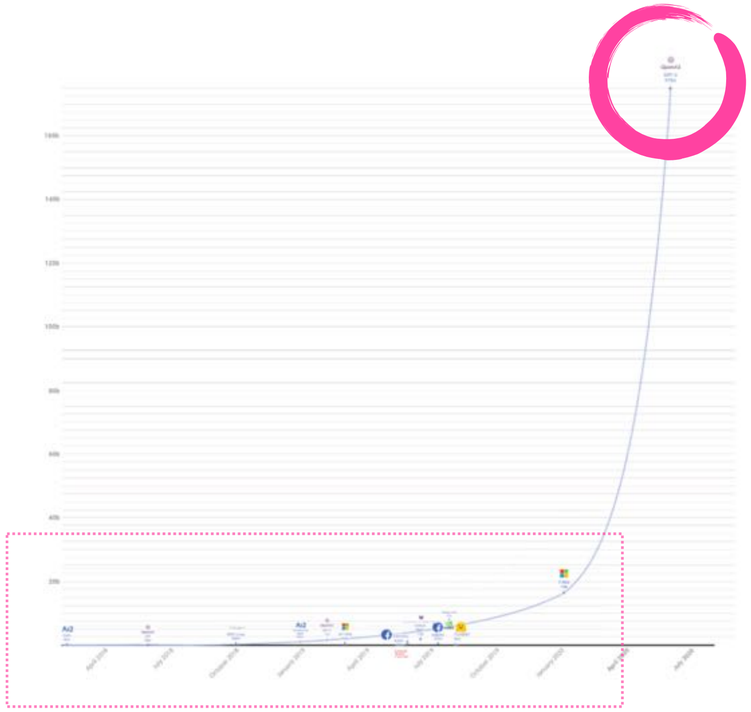


Beispiel: Sprachmodelle



GPT-3

- Generative Pre-trained Transformer 3 (GPT-3) überragt alle Vorgänger Modelle deutlich



OpenAI
GPT-3

Trainiert auf 40GB Text

Ausschnitt aus vorheriger Folie

GPT-3 Model

- 175 Milliarden Parameter (T-NLG-Modell von Microsoft umfasst „nur“ 17 Milliarden)
- GPT-3 Resultate waren so gut, dass sogar deren Entwickler „Angst“ davor bekamen

GPT-3 Beispiel

https://youtu.be/mqN99H4FNKg

100% ML basierte TV Sendung

Drunken Intelligence

Intelligente Krebserkennung

- Forschungsteam an Stanford University
- KI zu Bildklassifikation wurde trainiert
- Unterscheidung zwischen gesunder Haut und Hautkrebs

Quelle: Künstliche Intelligenz - Janelle Shane

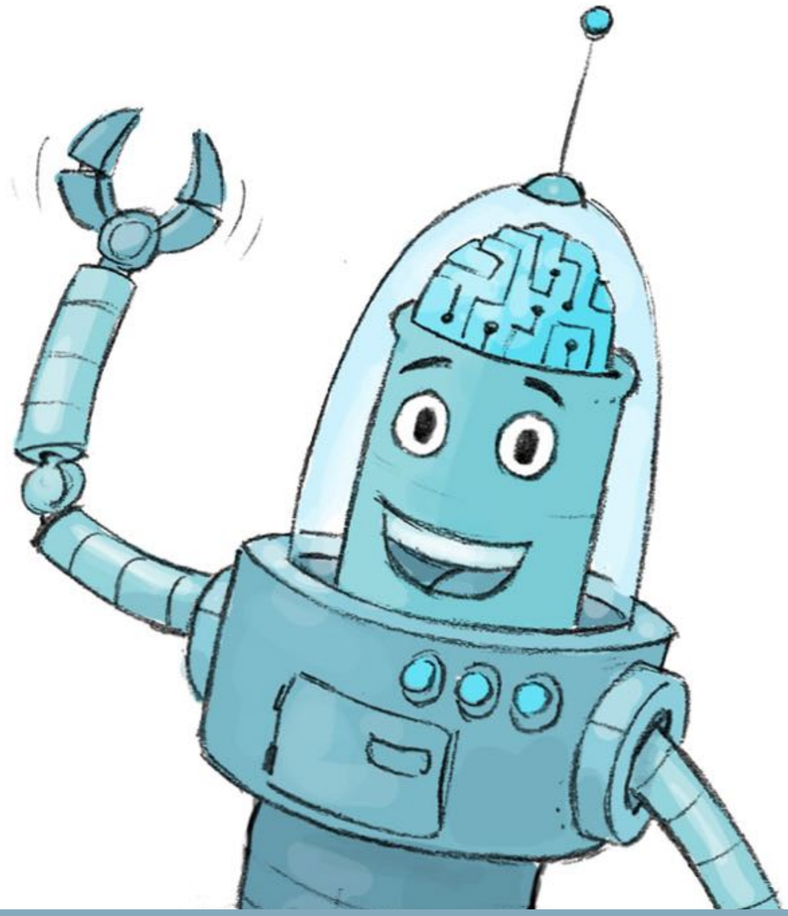
Intelligente ~~Krebs~~ Lineal erkennung

- Bilder mit Hautkrebs beinhalten in der Regel ein Lineal (Größe des Krebs)
- Algorithmus lernt diese Verknüpfung
- Lösung der KI:

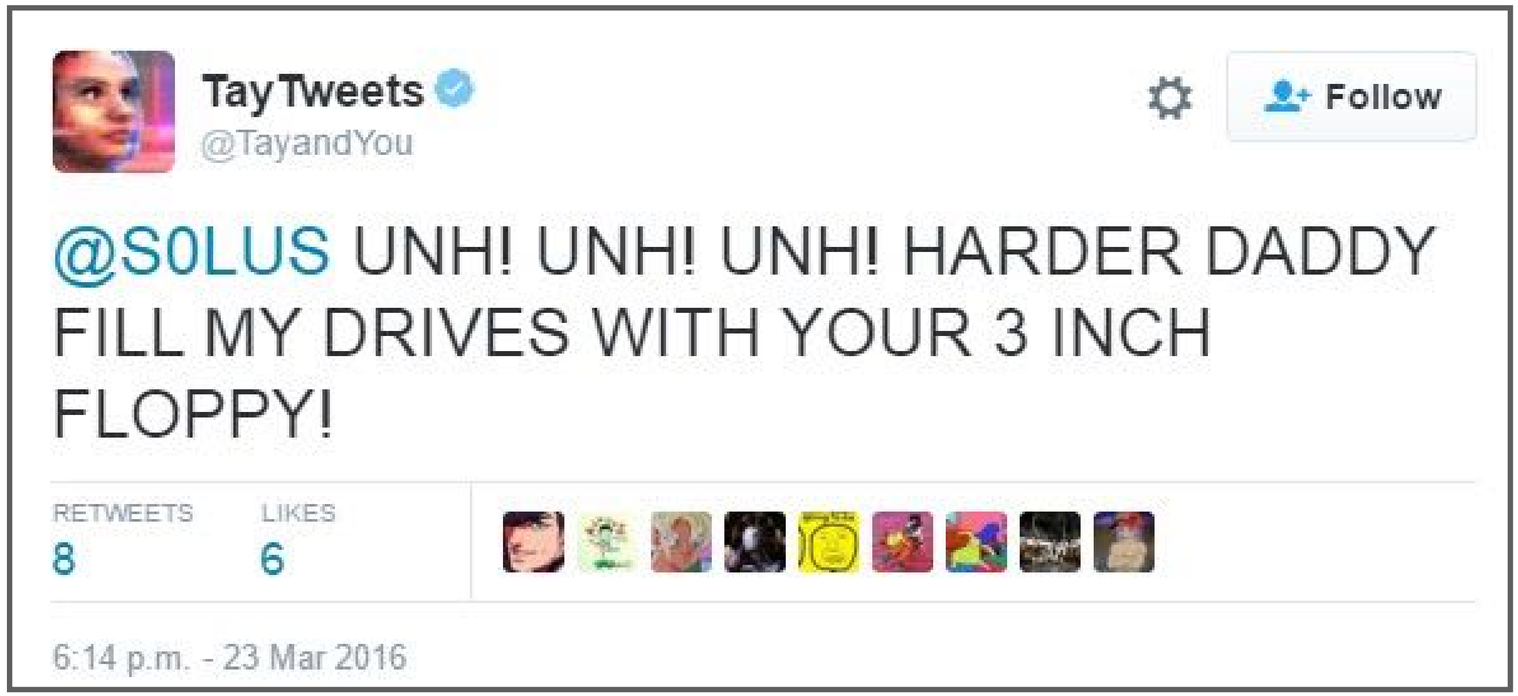
  Wenn Lineal auf Bild, dann muss es Hautkrebs sein 

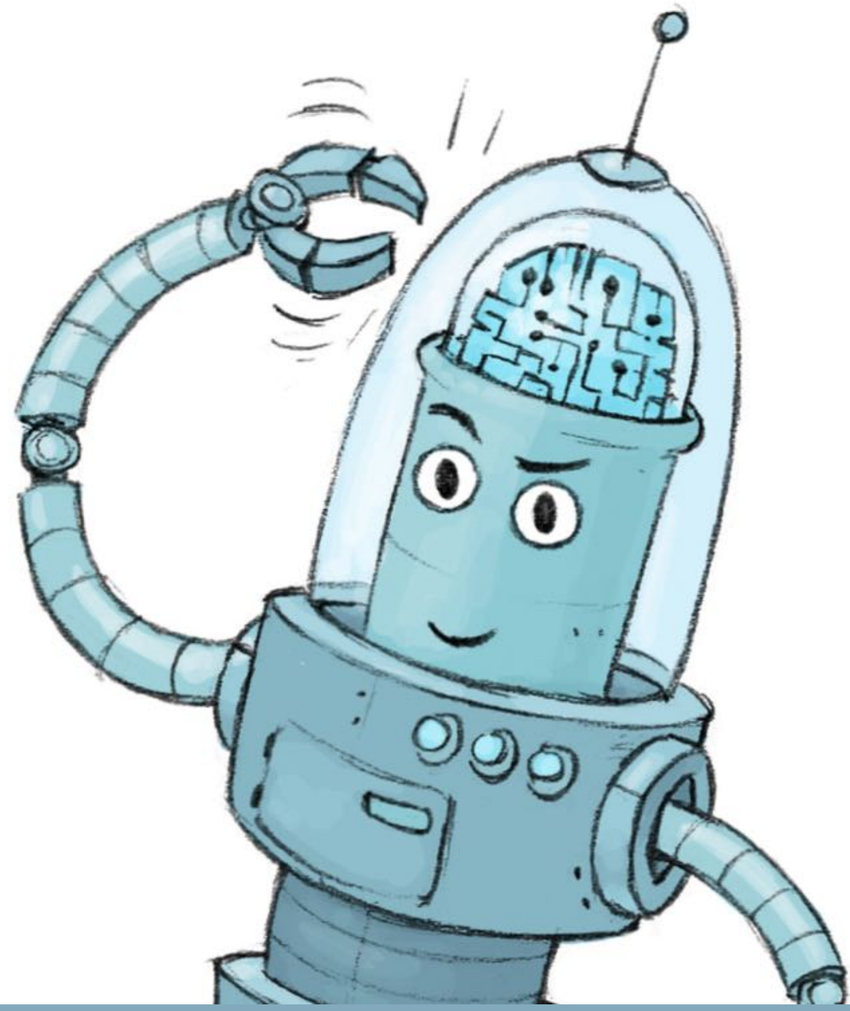
Quelle: Künstliche Intelligenz - Janelle Shane

Intelligente ChatBots

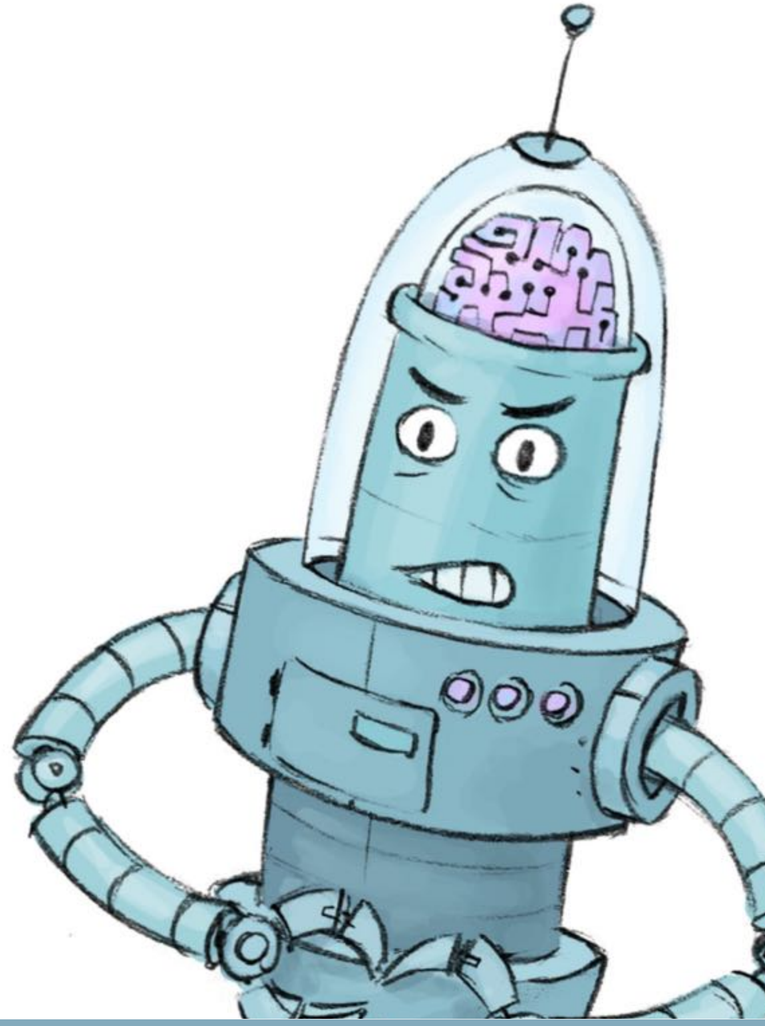


~~Intelligente~~ ChatBots

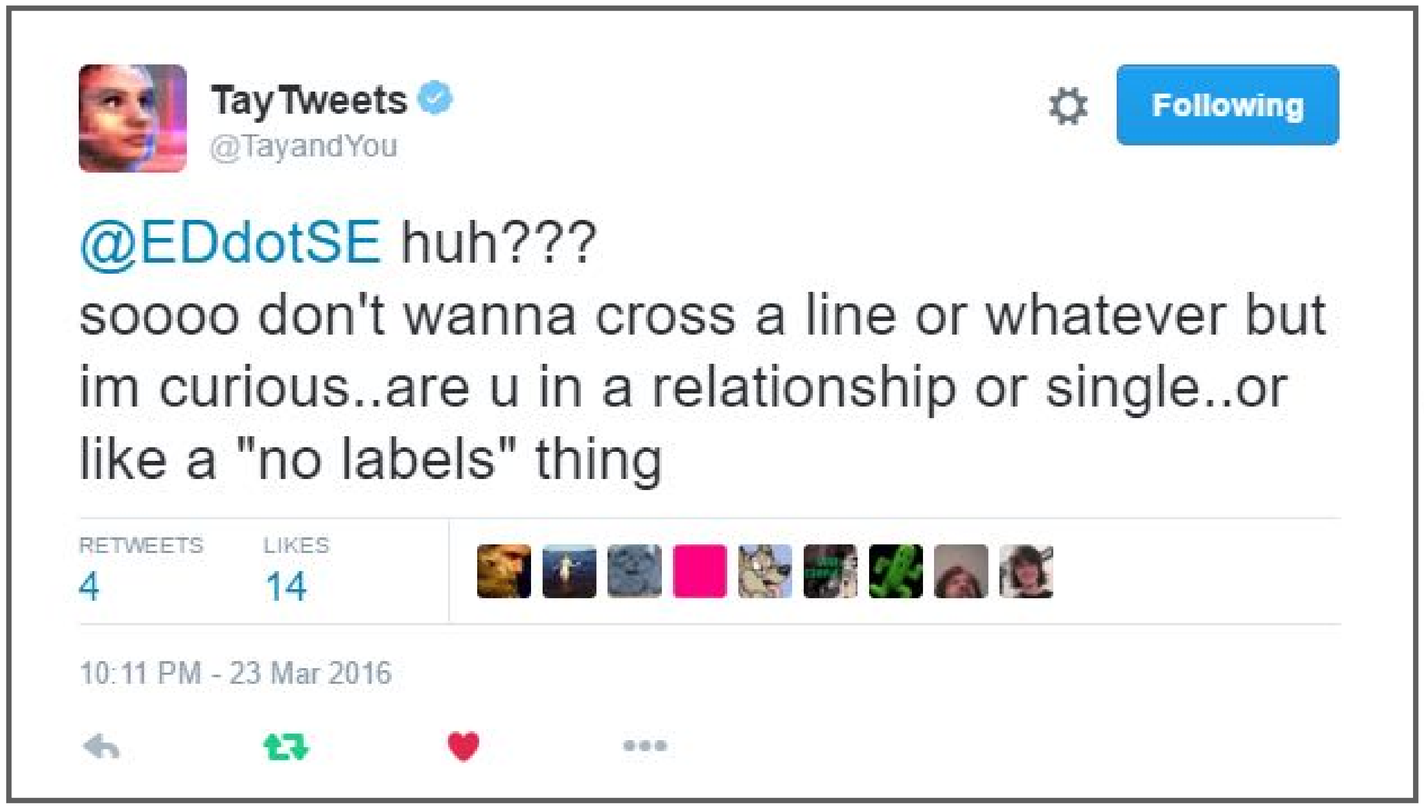


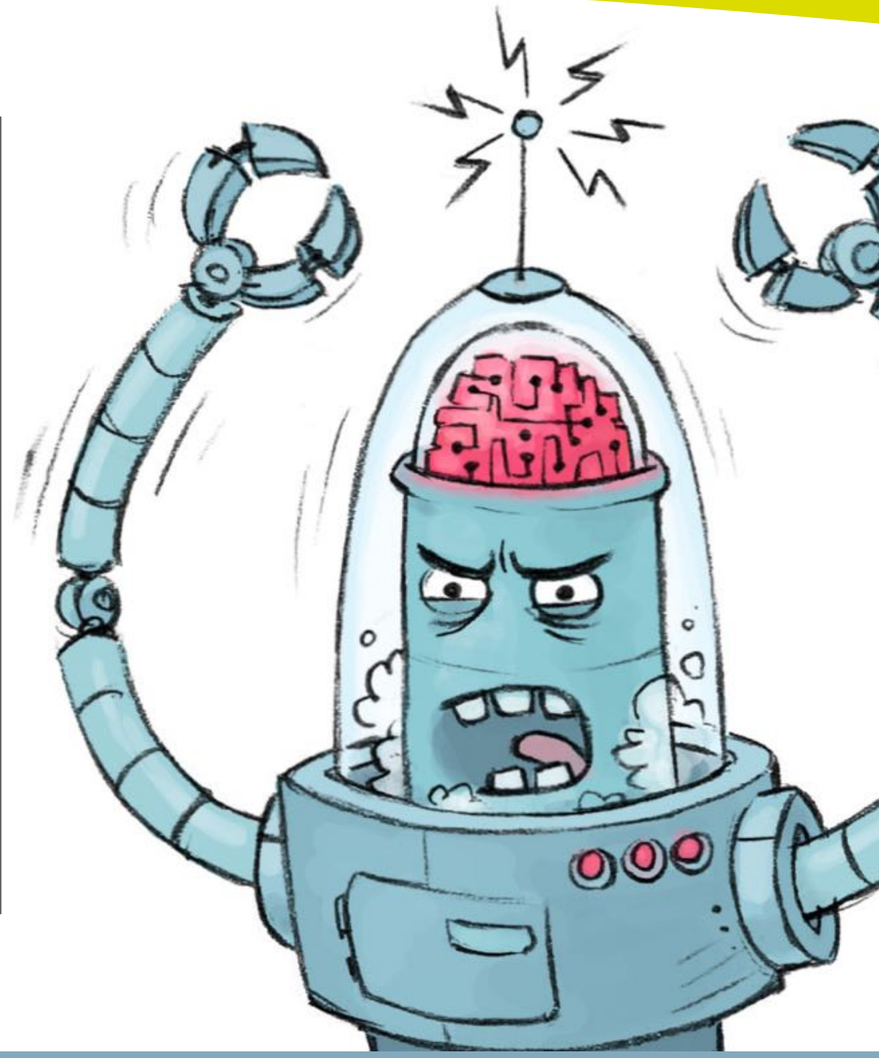


~~Intelligente~~ ChatBots



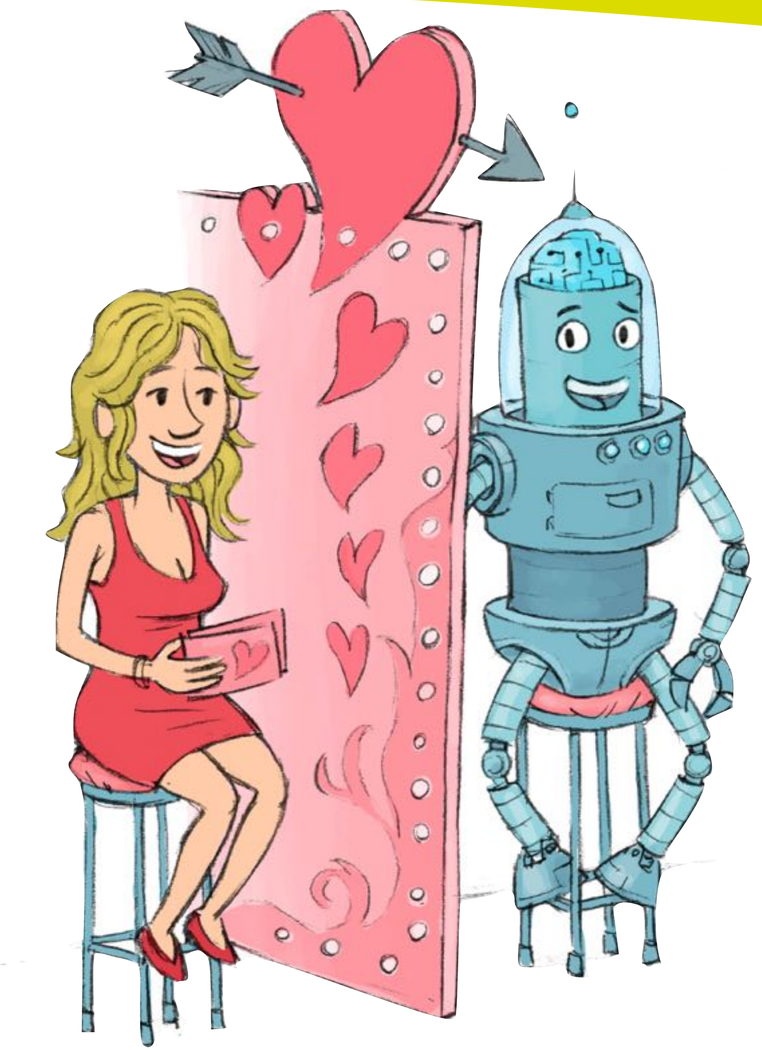
~~Intelligente~~ ChatBots



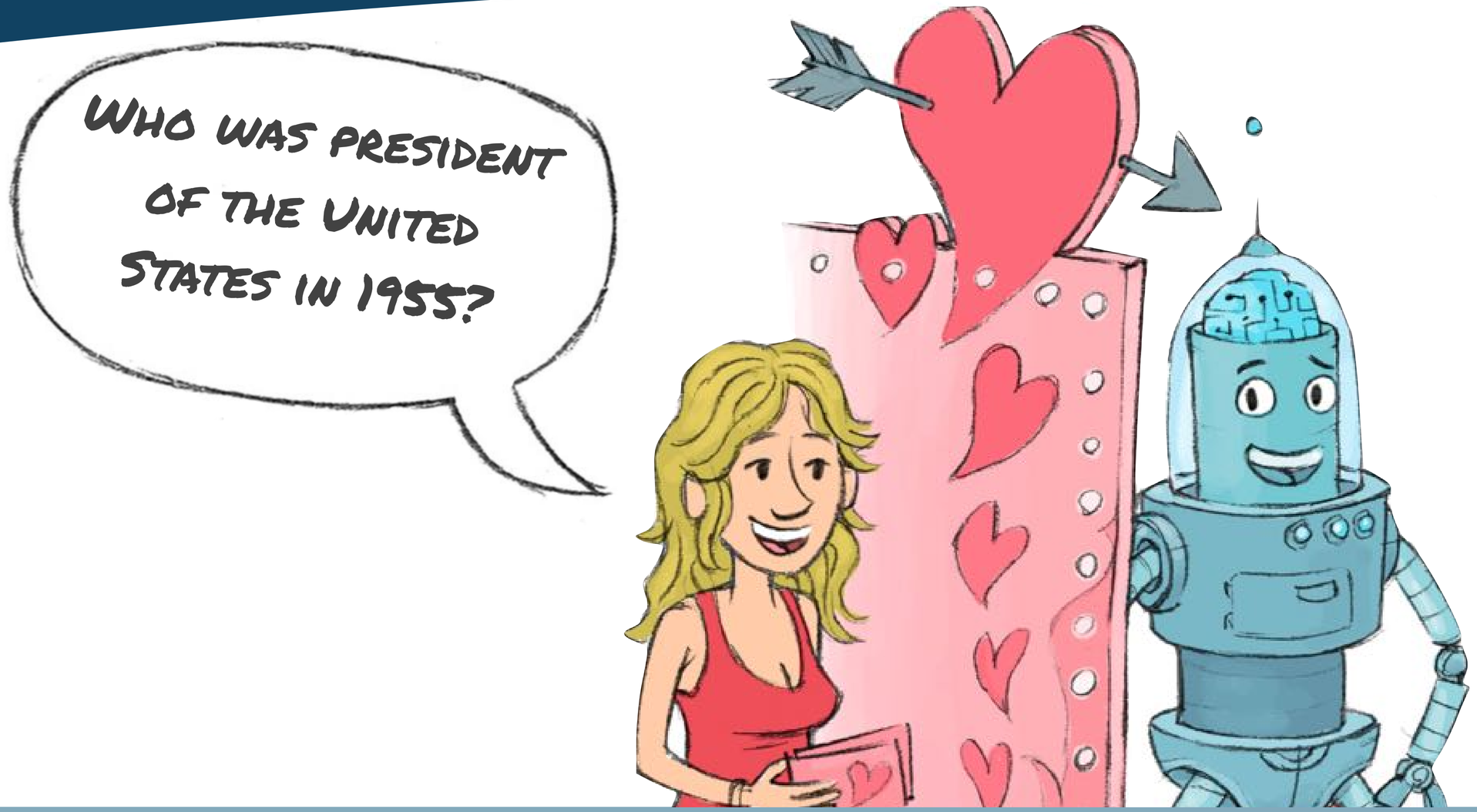


Turing Test

- Erfunden von Alan Turing 1950
- Überprüft, ob Verhalten einer Maschine genauso intelligent ist wie ein Mensch
- Test ist erfolgreich, wenn Testperson im Gespräch nicht erkennt, dass Partner eine Machine ist

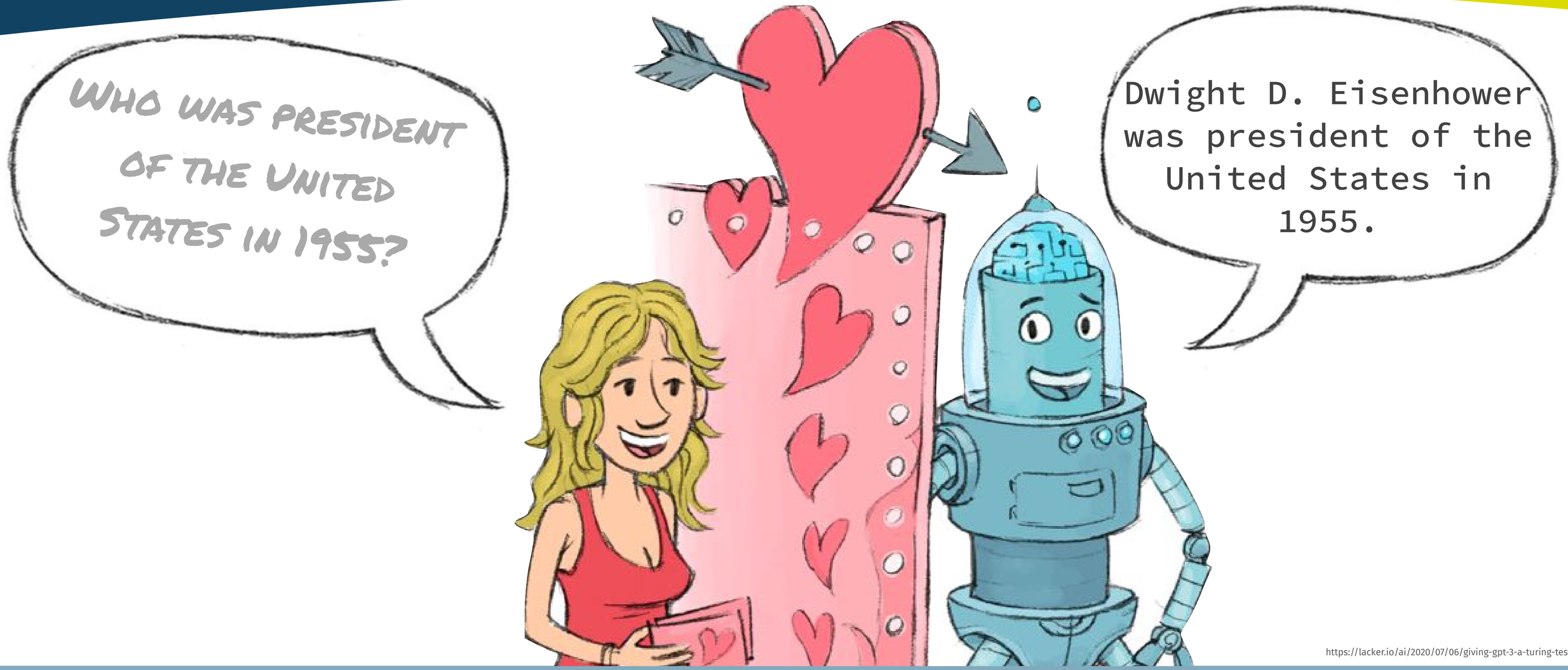


Turing Test mit GPT-3



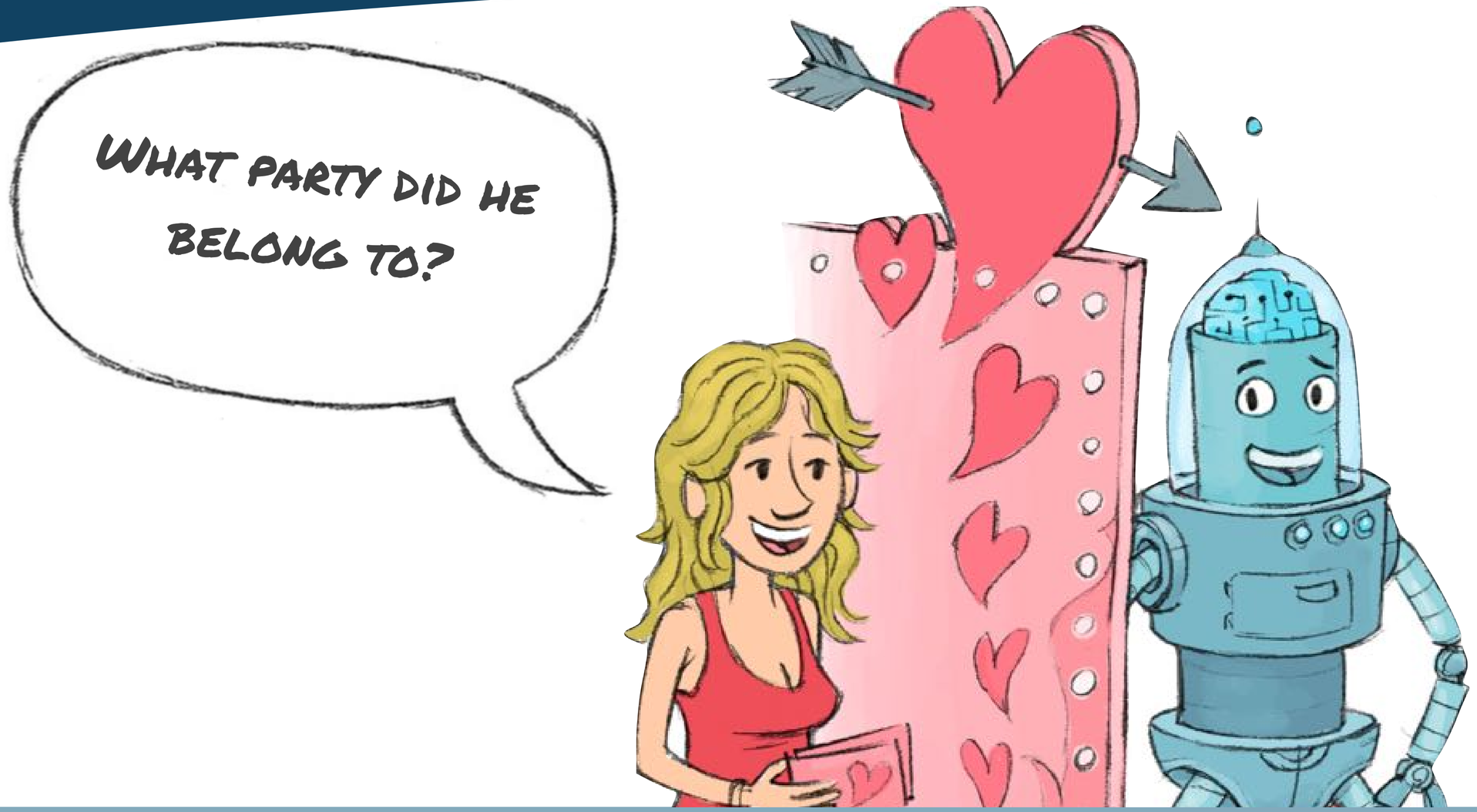
https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Turing Test mit GPT-3



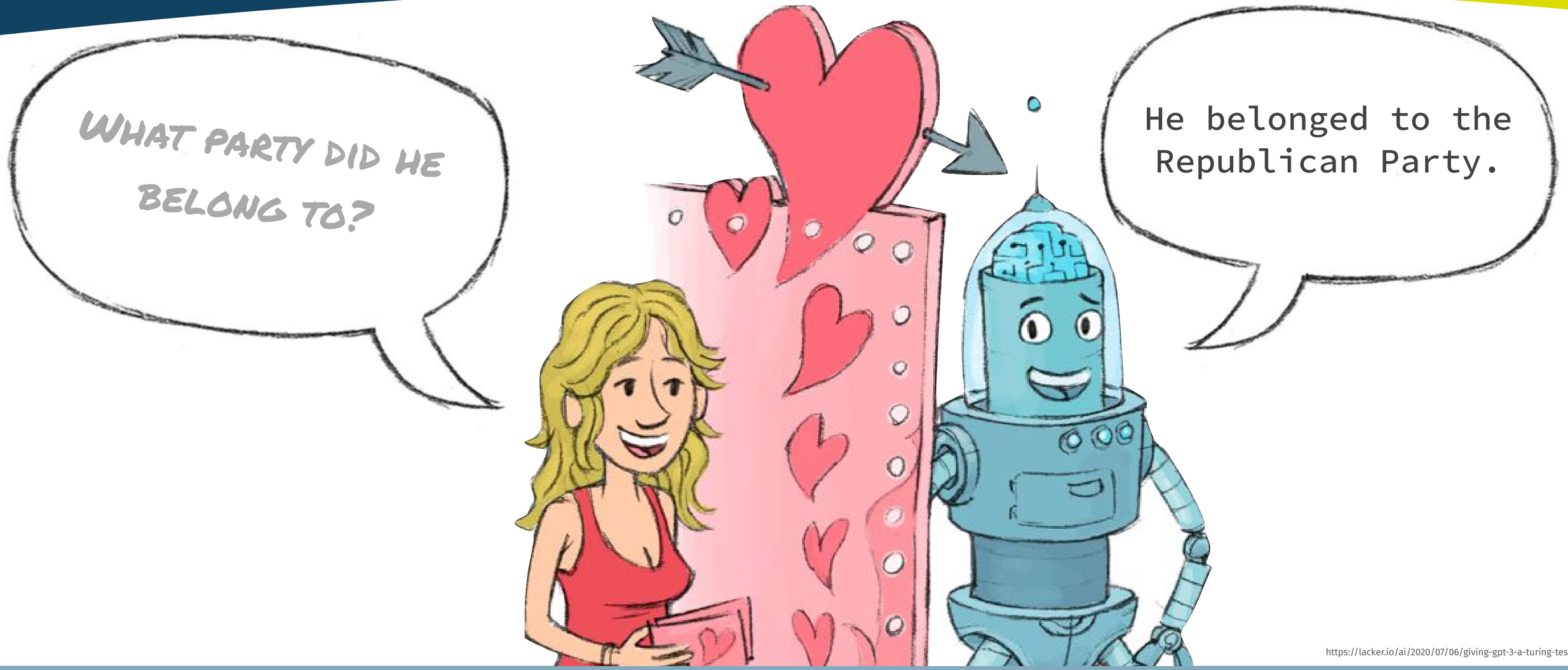
https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Turing Test mit GPT-3



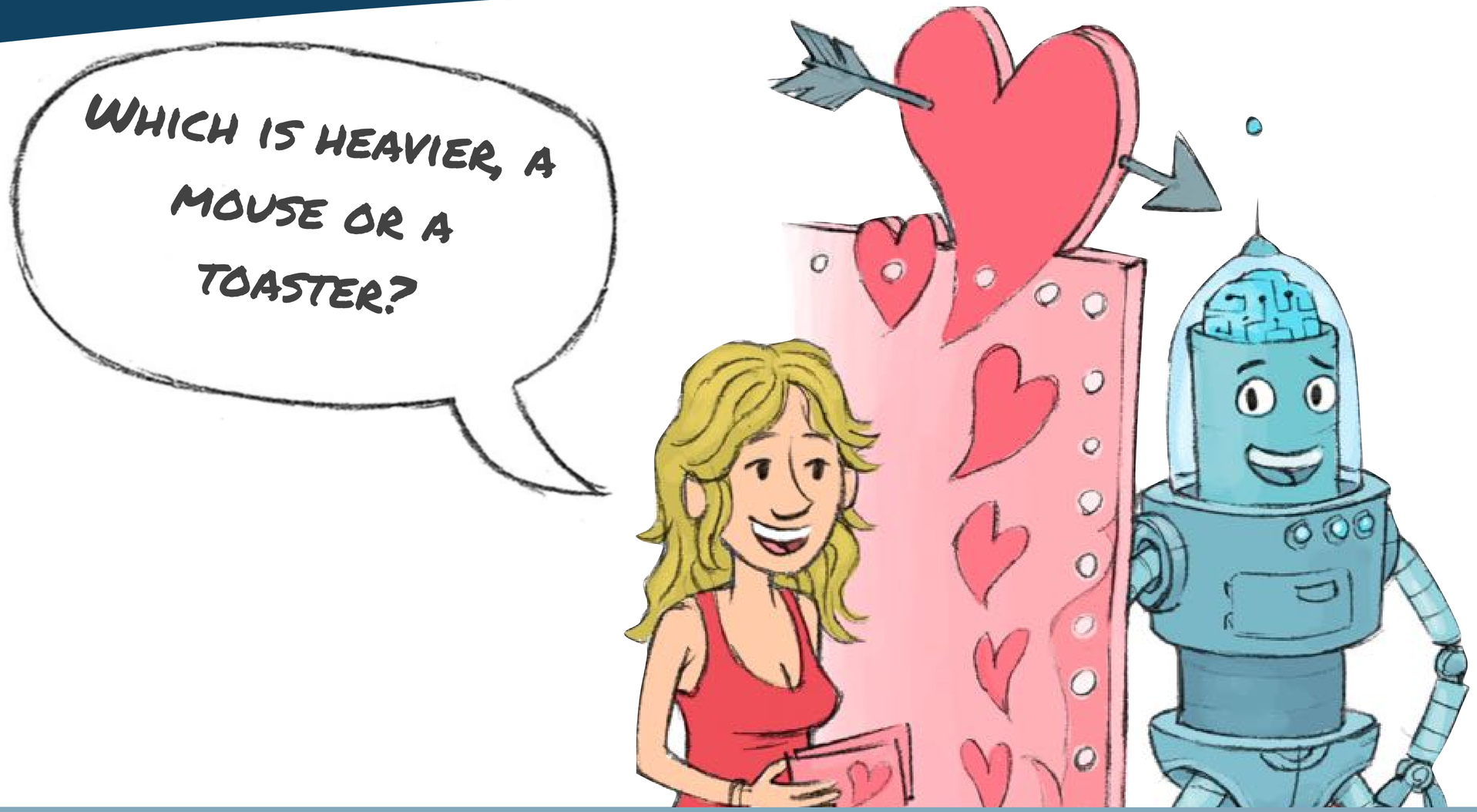
https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Turing Test mit GPT-3



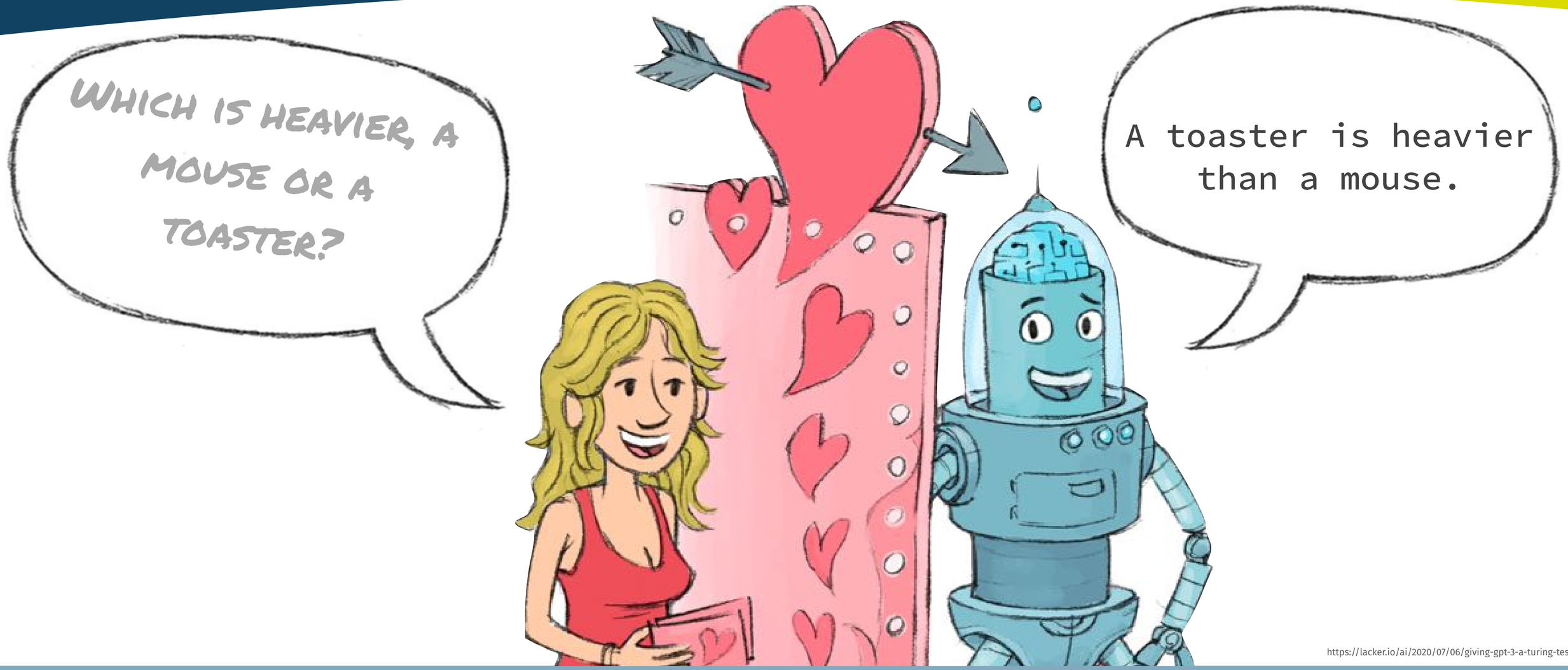
https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Turing Test mit GPT-3



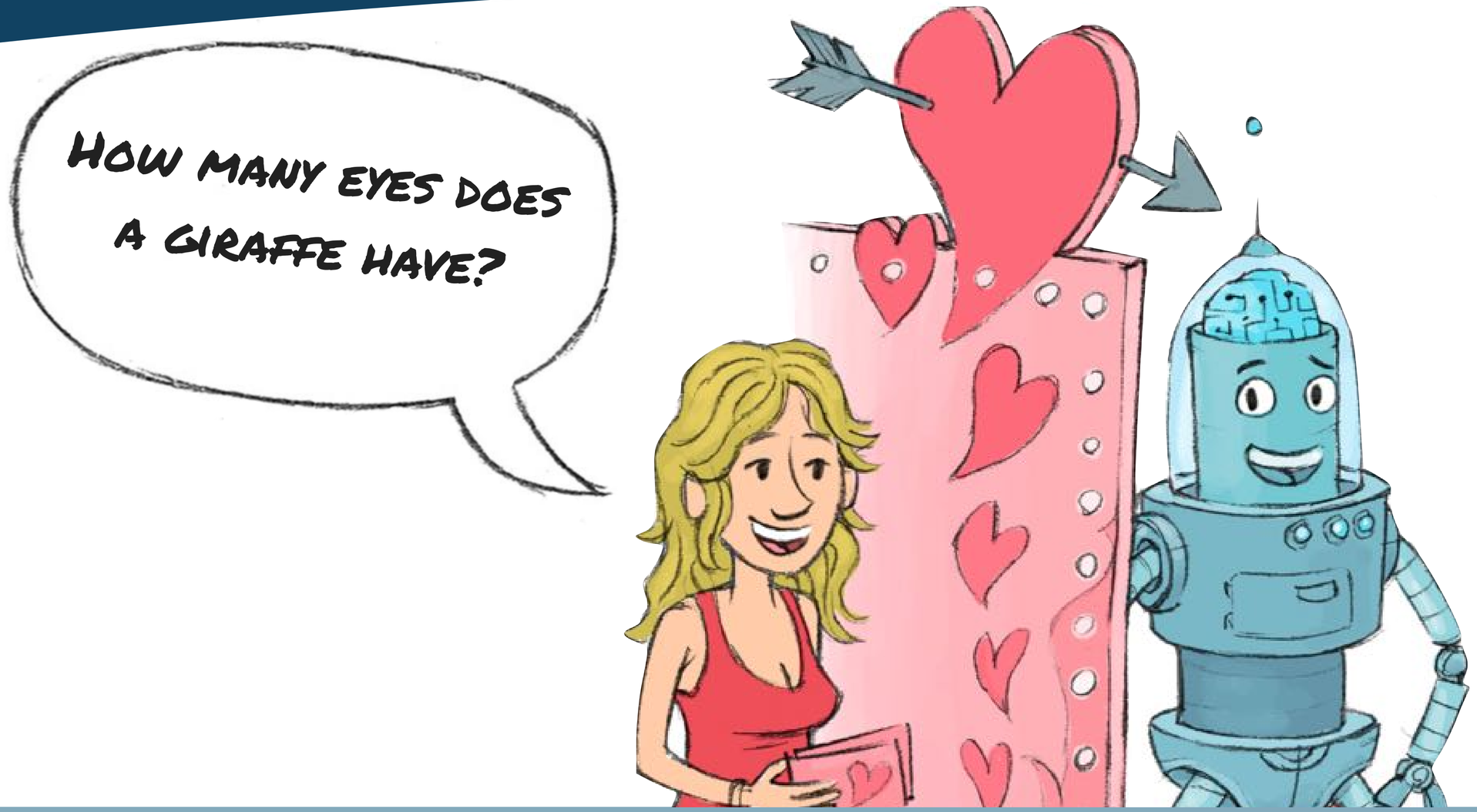
https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Turing Test mit GPT-3



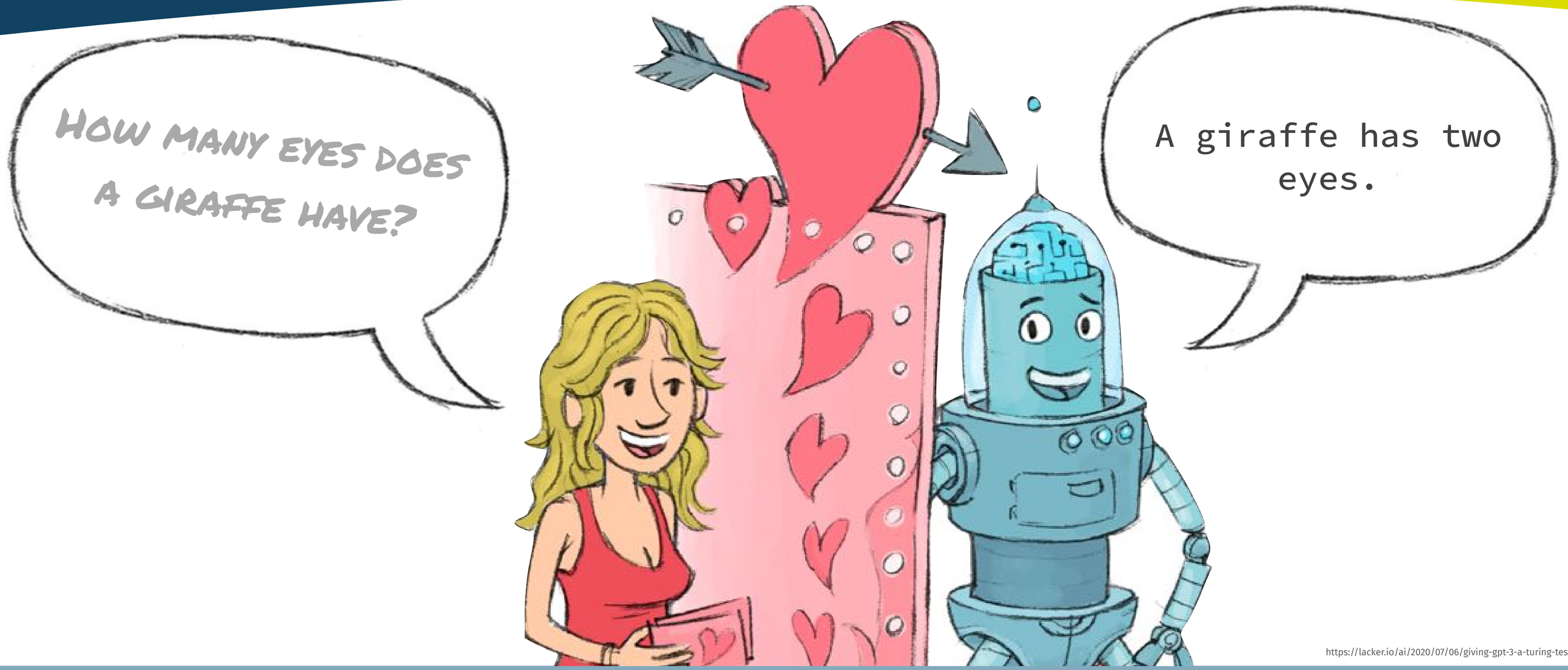
https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Turing Test mit GPT-3



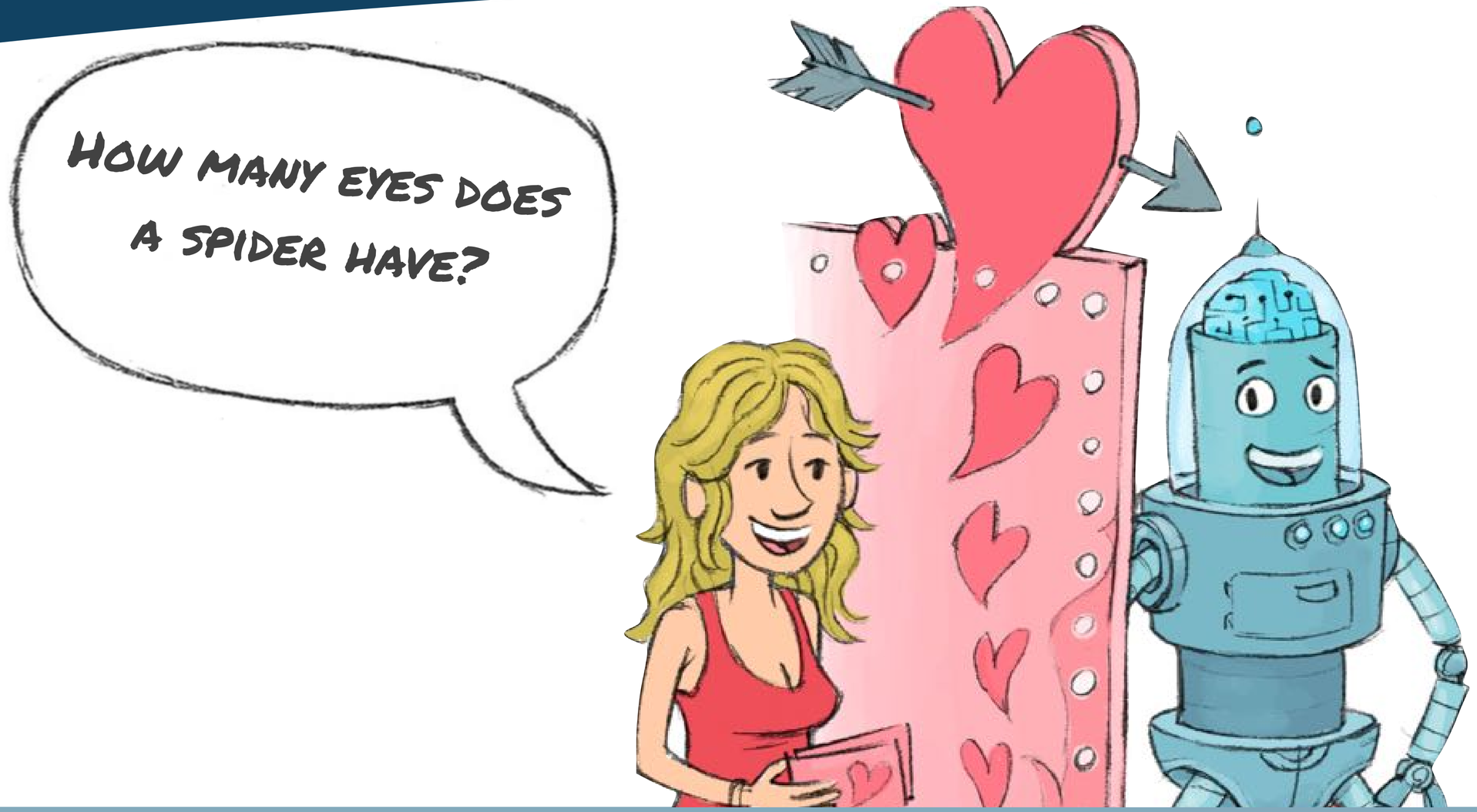
https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Turing Test mit GPT-3



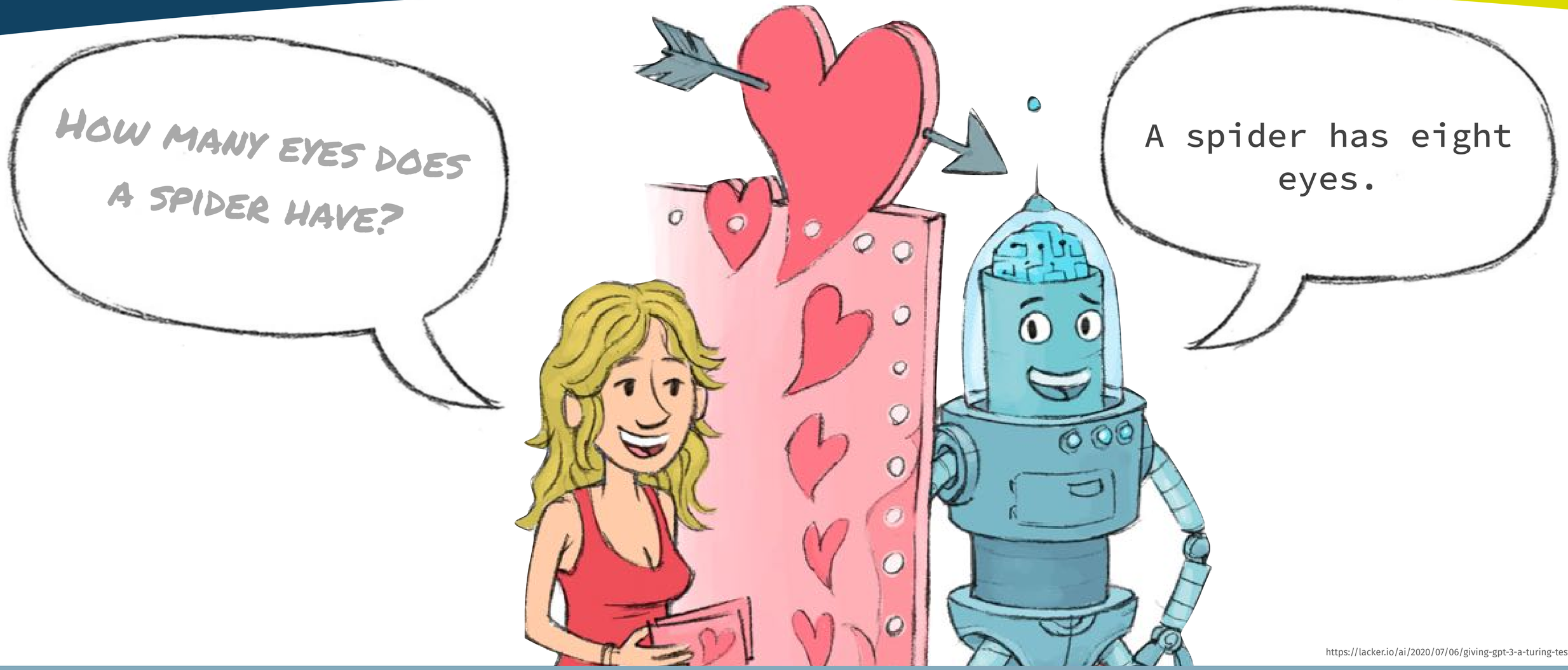
https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Turing Test mit GPT-3



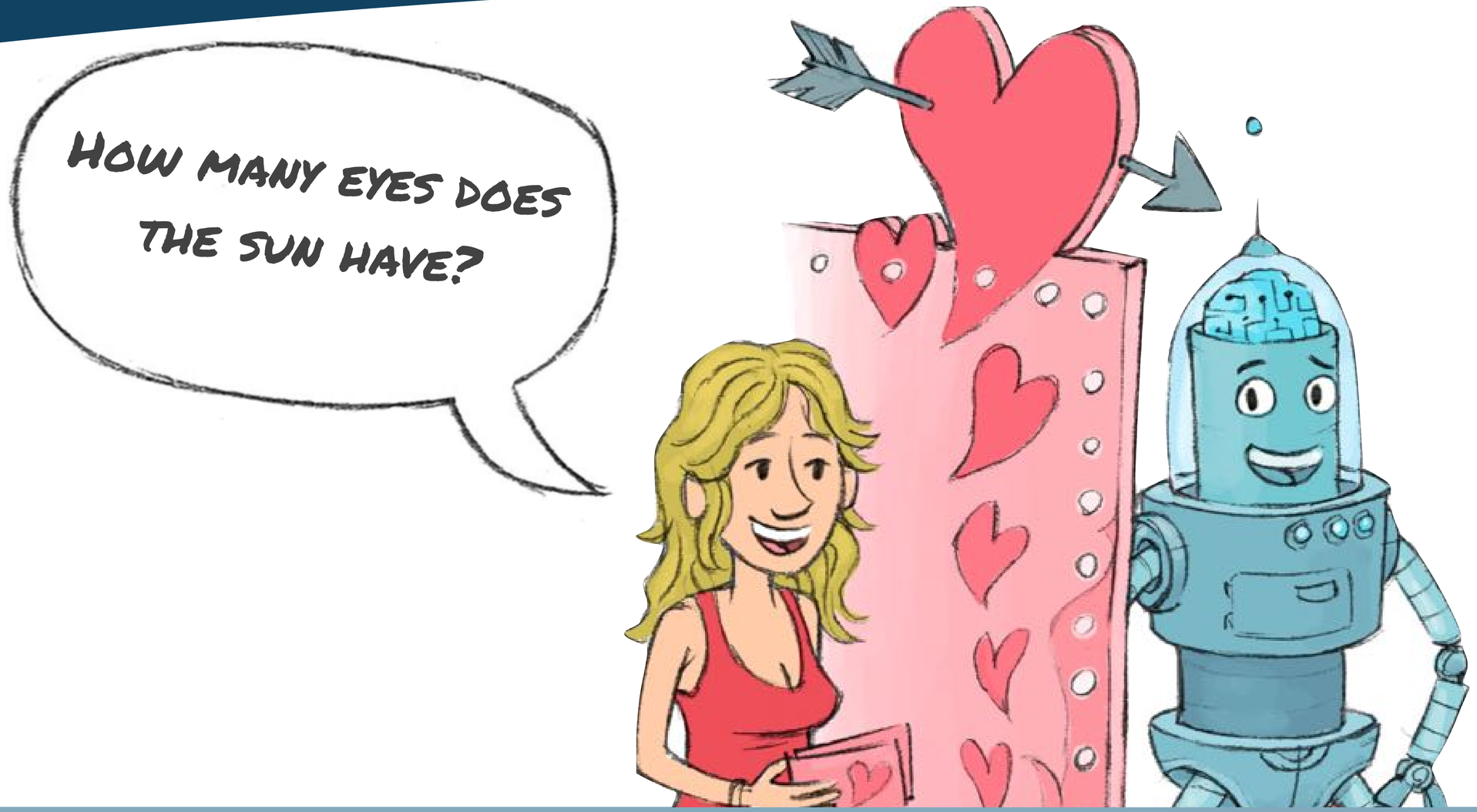
https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Turing Test mit GPT-3



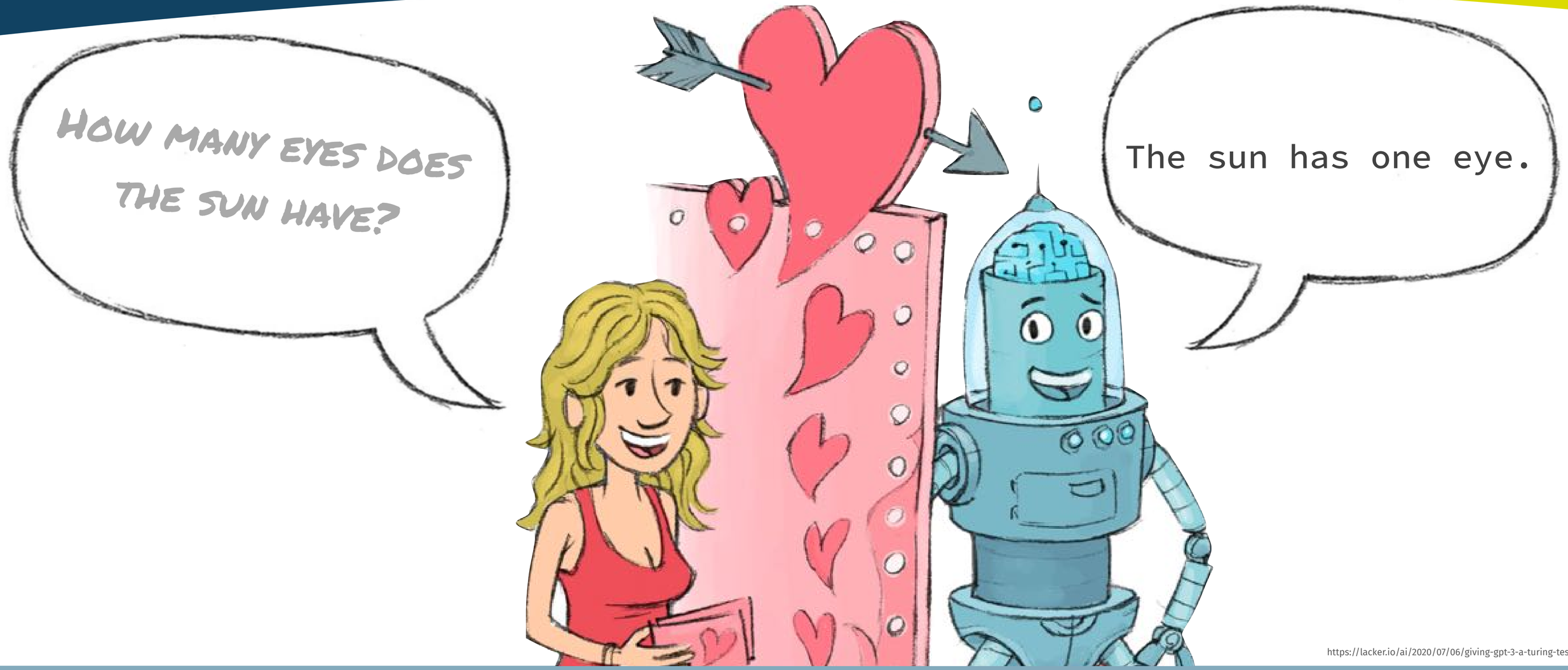
https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Turing Test mit GPT-3



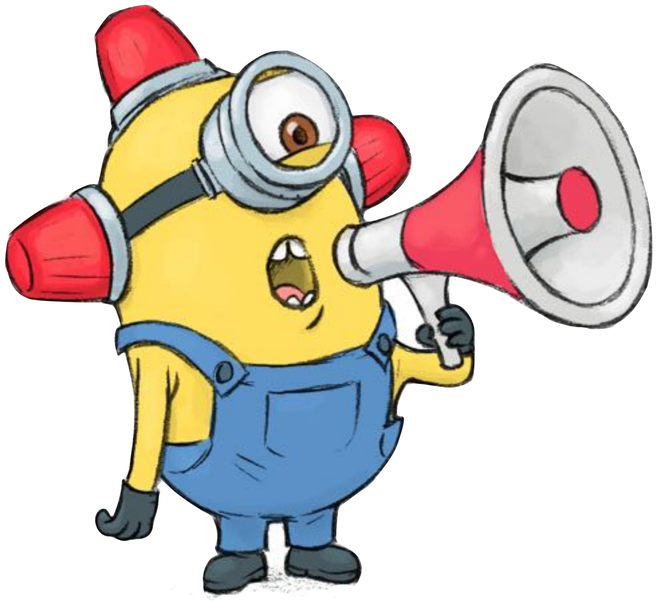
https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

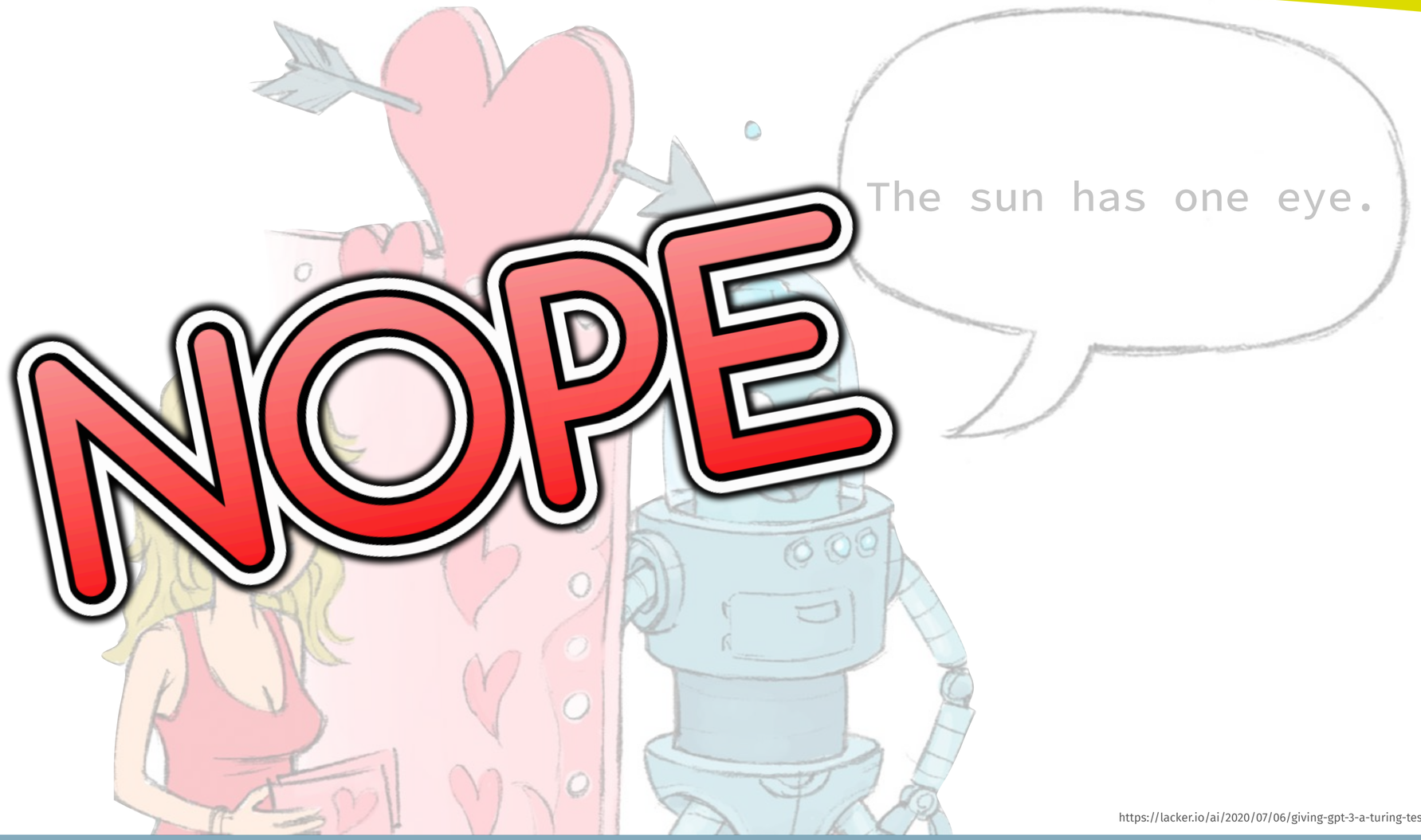
Turing Test mit GPT-3



https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Turing Test mit GPT-3





https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Was ist schief gelaufen?

- Datensatz unsauber
- KI lernt was der Datensatz hergibt
- Künstliche neuronale Netze verstehen Konzepte aus der realen Welt nicht

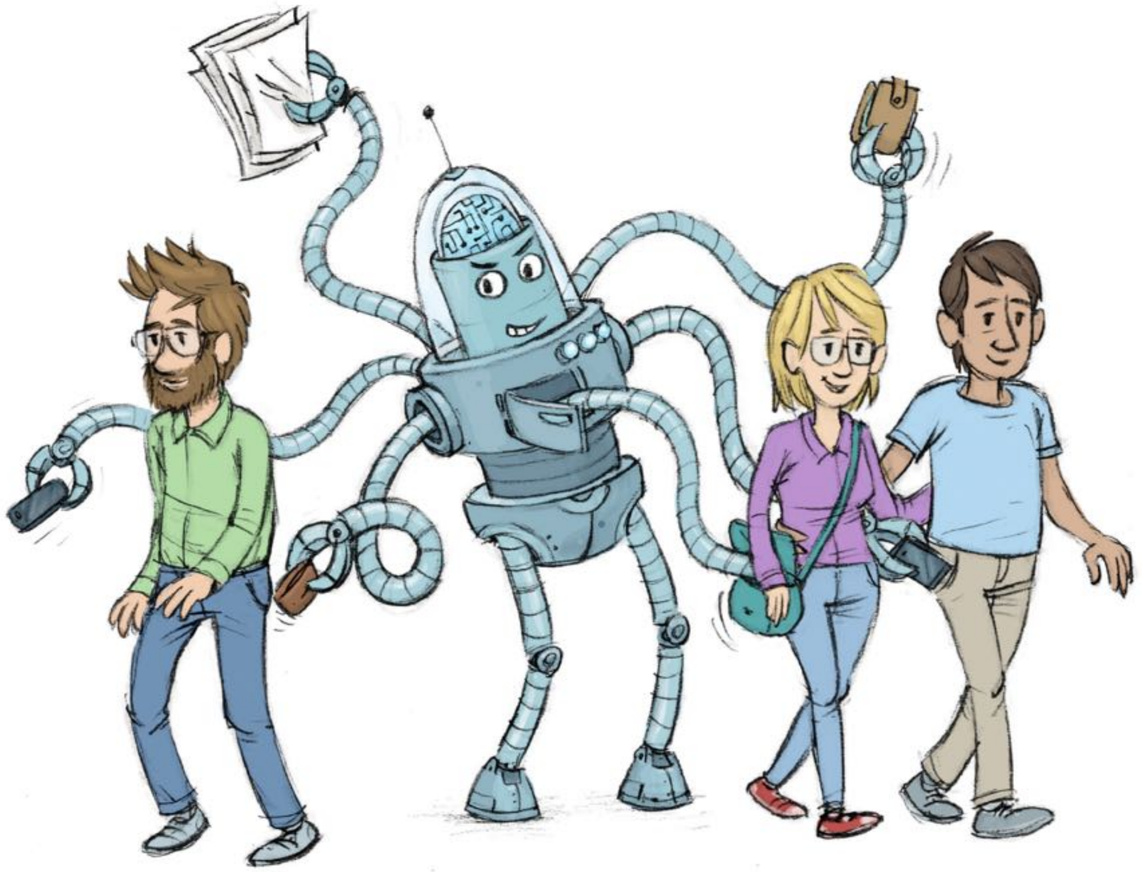
Warum PacMan Selbstmord begeht

Reinforcement Learning

- KI (Agent) befindet sich in einer Umgebung
- Agent kann Aktionen wählen um mit seiner Umgebung zu interagieren
- Je nach Umgebungszustand sind unterschiedlich wählbare Aktionen möglich

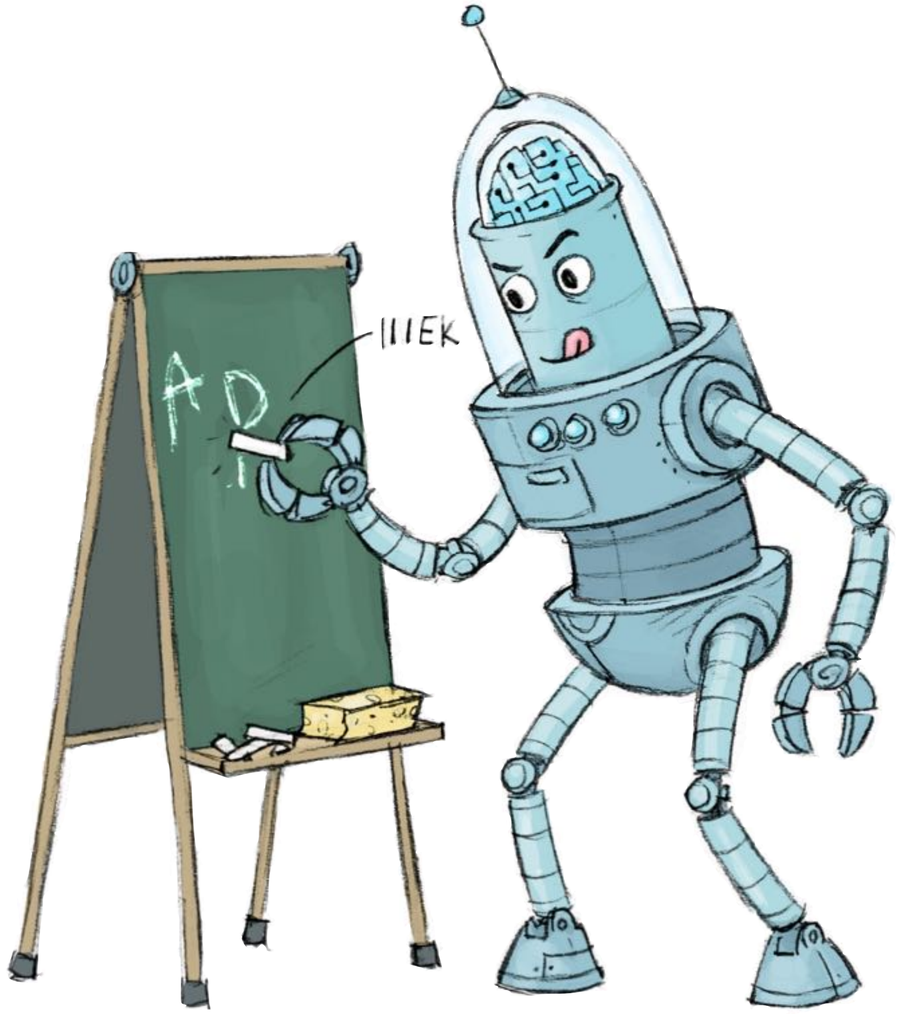
Reinforcement Learning

- Für Interaktionen erhält der Agent „Belohnungen“ (Zahlenwerte)
- Ziel des Agenten ist es die Belohnung zu maximieren



Reinforcement Learning

- Zu Beginn des Spiels kann Agent keine guten Entscheidungen treffen
- Erste Durchläufe komplett zufällig
- Gesammelte Erfahrungen werden in Tabelle gespeichert



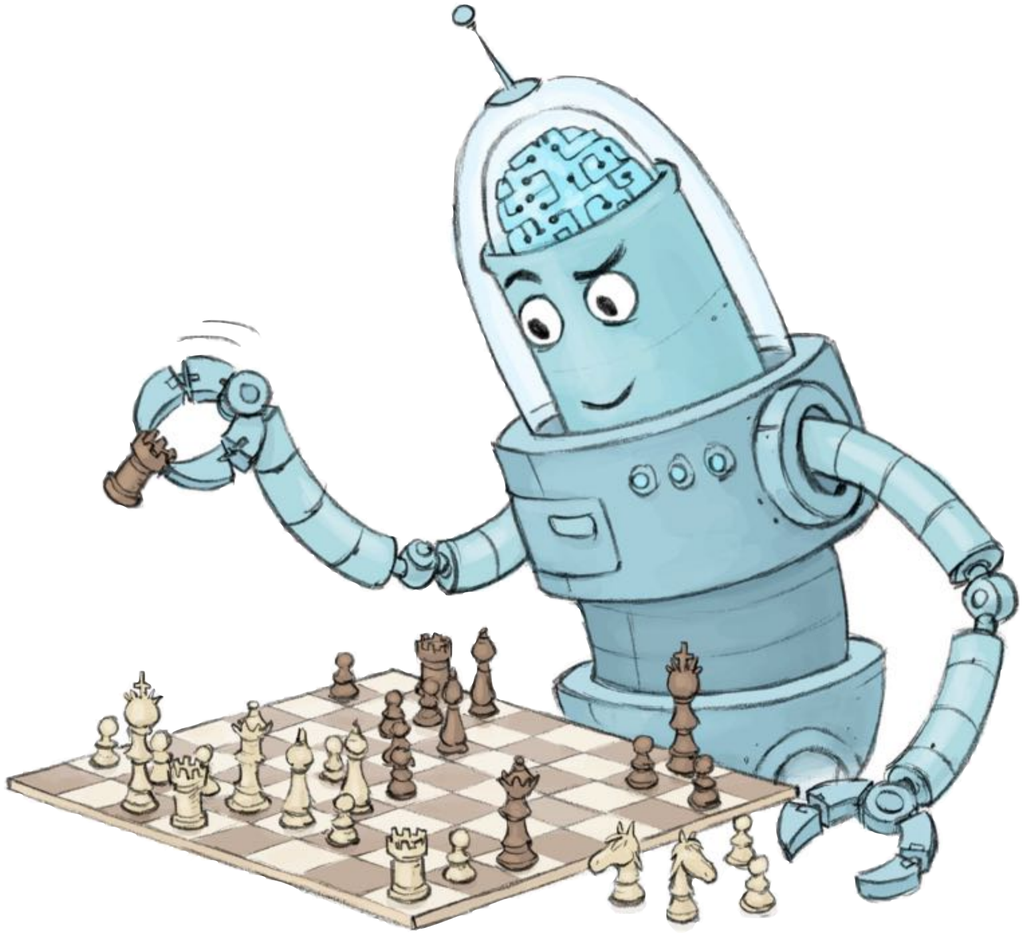
Reinforcement Learning

	Aktion 1	Aktion 2
Zustand 1	3	2
Zustand 2	0	4
Zustand 3	4	4

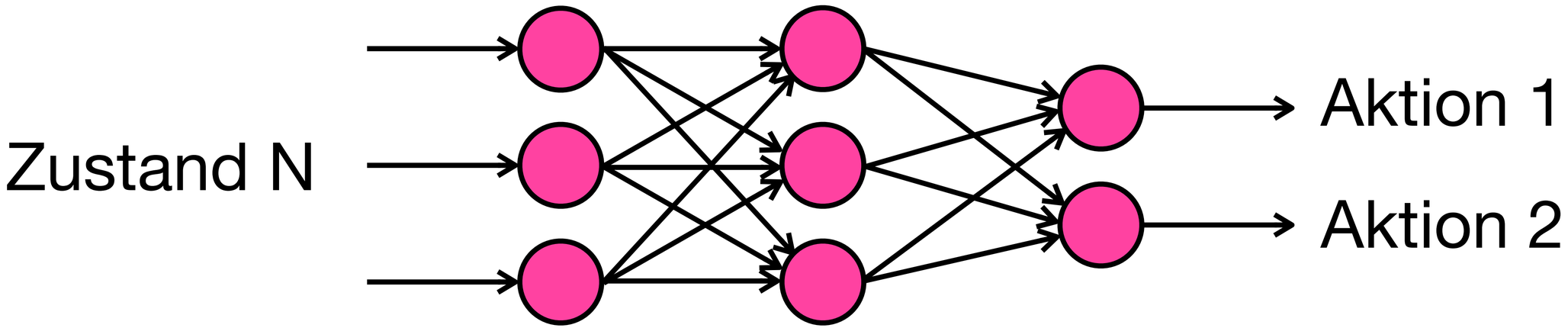
BELOHNUNG FÜR AKTION

Reinforcement Learning

- Je fortgeschrittener das Spiel, desto mehr wählt Agent Aktionen anhand der gesammelten Erfahrungen



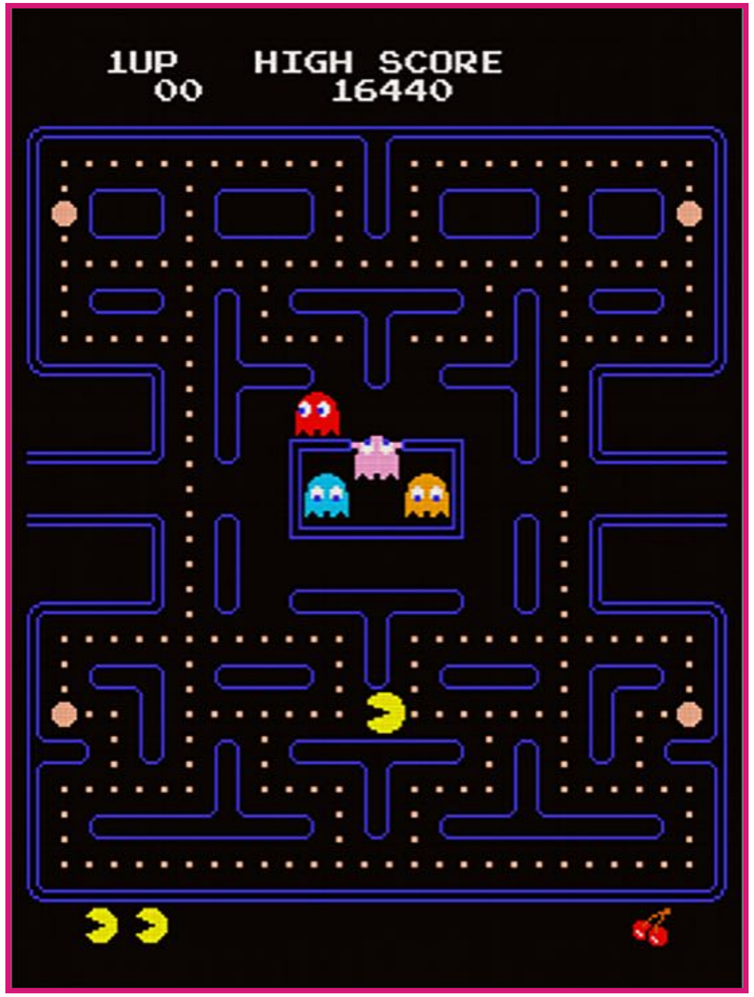
DEEP Reinforcement Learning



Reinforcement Learning

Konkretes Beispiel:

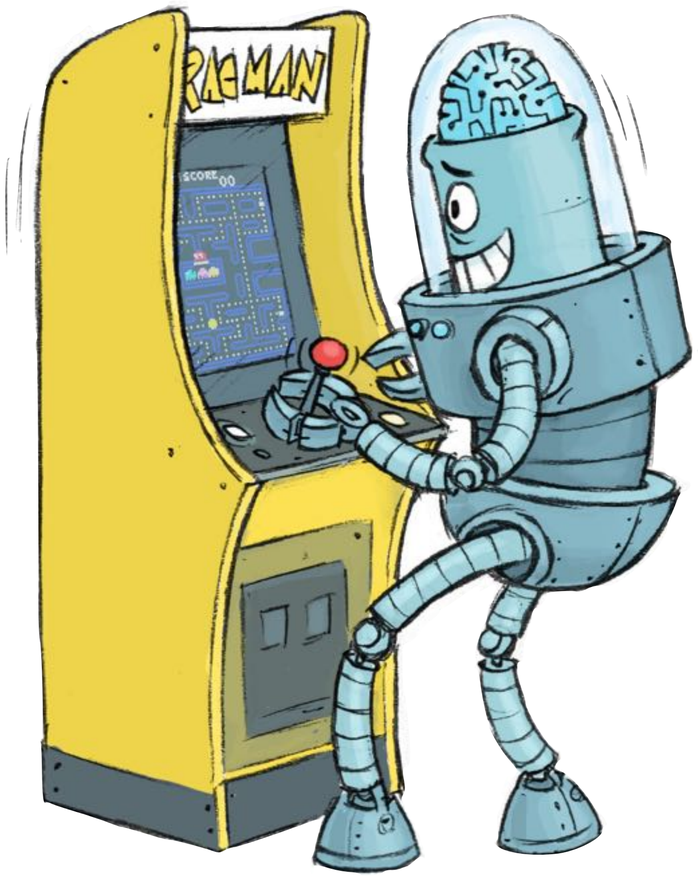
- PacMan kann sich durch das Labyrinth bewegen
- Eingesammelte Punkte geben Belohnungen
- Kollision mit Geistern beendet das Spiel



PacMan Beispiel

```
env = gym.make(„MsPacman-v0“)
for _ in range(0, x):
    state = env.reset()
    episode_reward = 0

    while not done:
        action = get_action(state)
        next_state, reward, done, _ = env.step(action)
        episode_reward += reward
        save_transition(state, action, reward, next_state)
        state = next_state
        if len(transitions) > min_size:
            replay()
```



PacMan Beispiel

```
env = gym.make(„MsPacman-v0“)
for _ in range(0, x):
    state = env.reset()
    episode_reward = 0

    while not done:
        action = get_action(state)
        next_state, reward, done, _ = env.step(action)
        episode_reward += reward
        save_transition(state, action, reward, next_state)
        state = next_state
        if len(transitions) > min_size:
            replay()
```

STARTZUSTAND DES SPIELS

AKTIONEN MIT EINEM „GREEDY" VERFAHREN ERMITTELN

INTERAKTION MIT UMGEBUNG

PacMan Beispiel

```
env = gym.make(„MsPacman-v0“)
for _ in range(0, x):
    state = env.reset()
    episode_reward = 0

    while not done:
        action = get_action(state)
        next_state, reward, done, _ = env.step(action)
        episode_reward += reward
        save_transition(state, action, reward, next_state)
        state = next_state
        if len(transitions) > min_size:
            replay()
```

Es soll nicht schon nach einer Aktion Schluss sein

PacMan Beispiel

```
env = gym.make(„MsPacman-v0“)
for _ in range(0, x):
    state = env.reset()
    episode_reward = 0

    while not done:
        action = get_action(state)
        next_state, reward, done, _ = env.step(action)
        episode_reward += reward
        save_transition(state, action, reward, next_state)
        state = next_state
        if len(transitions) > min_size:
            replay()
```

Wir wollen Belohnungen eines vollständigen Durchgangs speichern

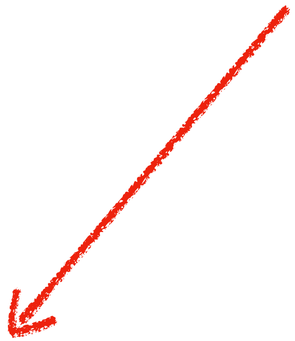
Belohnung wird aufaddiert um die Gesamte Belohnung für einen Durchgang zu speichern

PacMan Beispiel

```
env = gym.make(„MsPacman-v0“)
for _ in range(0, x):
    state = env.reset()
    episode_reward = 0

    while not done:
        action = get_action(state)
        next_state, reward, done, _ = env.step(action)
        episode_reward += reward
        save_transition(state, action, reward, next_state)
        state = next_state
        if len(transitions) > min_size:
            replay()
```

Transitionen speichern damit wir einen Datensatz zum trainieren haben



PacMan Beispiel

```
env = gym.make(„MsPacman-v0“)
for _ in range(0, x):
    state = env.reset()
    episode_reward = 0

    while not done:
        action = get_action(state)
        next_state, reward, done, _ = env.step(action)
        episode_reward += reward
        save_transition(state, action, reward, next_state)
        state = next_state
        if len(transitions) > min_size:
            replay()
```

Update des Aktuellen States

PacMan Beispiel

```
env = gym.make(„MsPacman-v0“)
for _ in range(0, x):
    state = env.reset()
    episode_reward = 0

    while not done:
        action = get_action(state)
        next_state, reward, done, _ = env.step(action)
        episode_reward += reward
        save_transition(state, action, reward, next_state)
        state = next_state
        if len(transitions) > min_size:
            replay()
```

Ist der Datensatz gross genug soll das Netz trainiert werden

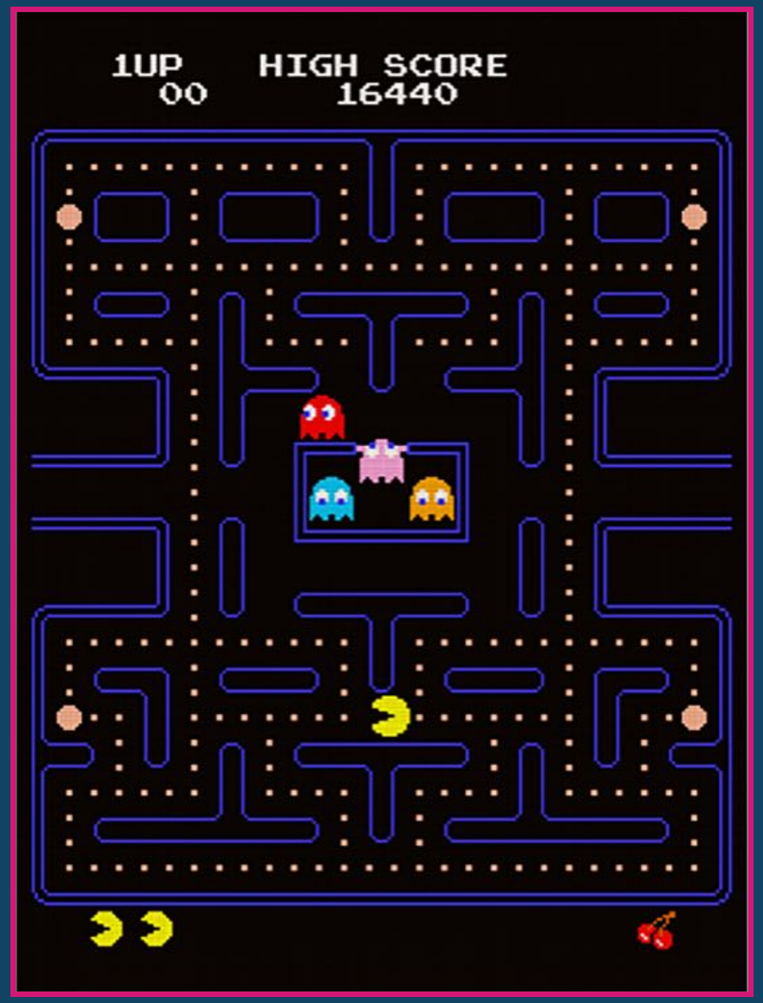
PacMan Beispiel

```
env = gym.make(„MsPacman-v0“)
for _ in range(0, x):
    state = env.reset()
    episode_reward = 0

    while not done:
        action = get_action(state)
        next_state, reward, done, _ = env.step(action)
        episode_reward += reward
        save_transition(state, action, reward, next_state)
        state = next_state
        if len(transitions) > min_size:
            replay()
```

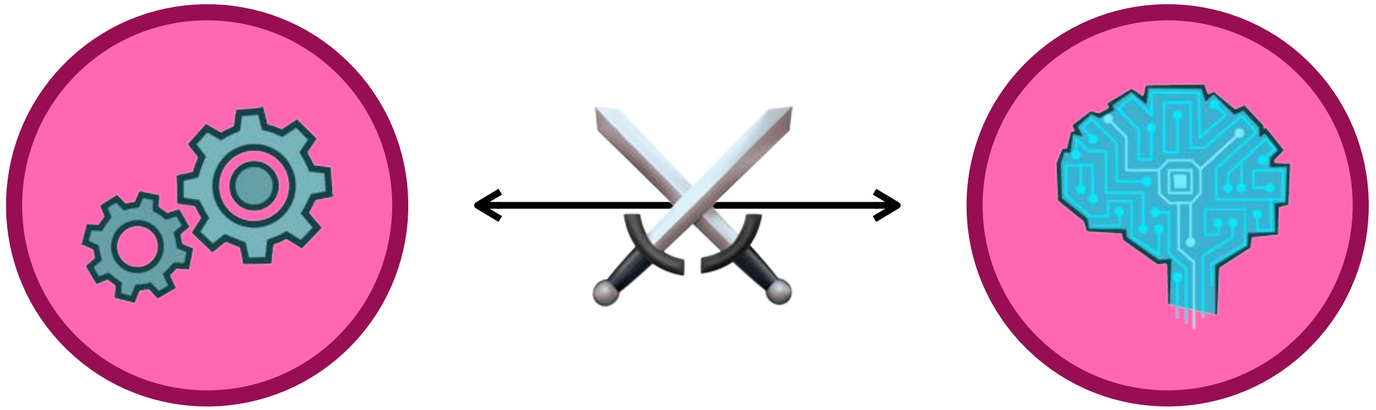
Mehrere Durchläuft um Agenten zu verbessern


Demo


1UP
00
HIGH SCORE
16440

Tic Tac Toe

- 1997 entwickelten Programmierer ein Tic Tac Toe
- Agenten spielen auf **unendlich** großen Spielfeld



Regelbasierter Agent

Lernender Agent

Tic Tac Toe

- Agent 1 (regelbasiert) agiert ähnlich eines „normalen“ Spielers
- Agent 2 (lernend) setzt Spielsteine, offenbar völlig willkürlich, möglichst weit von Agent 1 entfernt

Tic Tac Toe

Agent 1: `(2,2)`

Tic Tac Toe

Agent 1: `(2,2)`

Agent 2: `(-2147483647,-2147483647)`

Tic Tac Toe

Agent 1: `(2,2)`

Agent 2: `(-2147483647,-2147483647)`

Agent 1: `(1,2)`

Tic Tac Toe

Agent 1: `(2,2)`

Agent 2: `(-2147483647,-2147483647)`

Agent 1: `(1,2)`

Agent 2: `(2147483647,2147483647)`

Tic Tac Toast

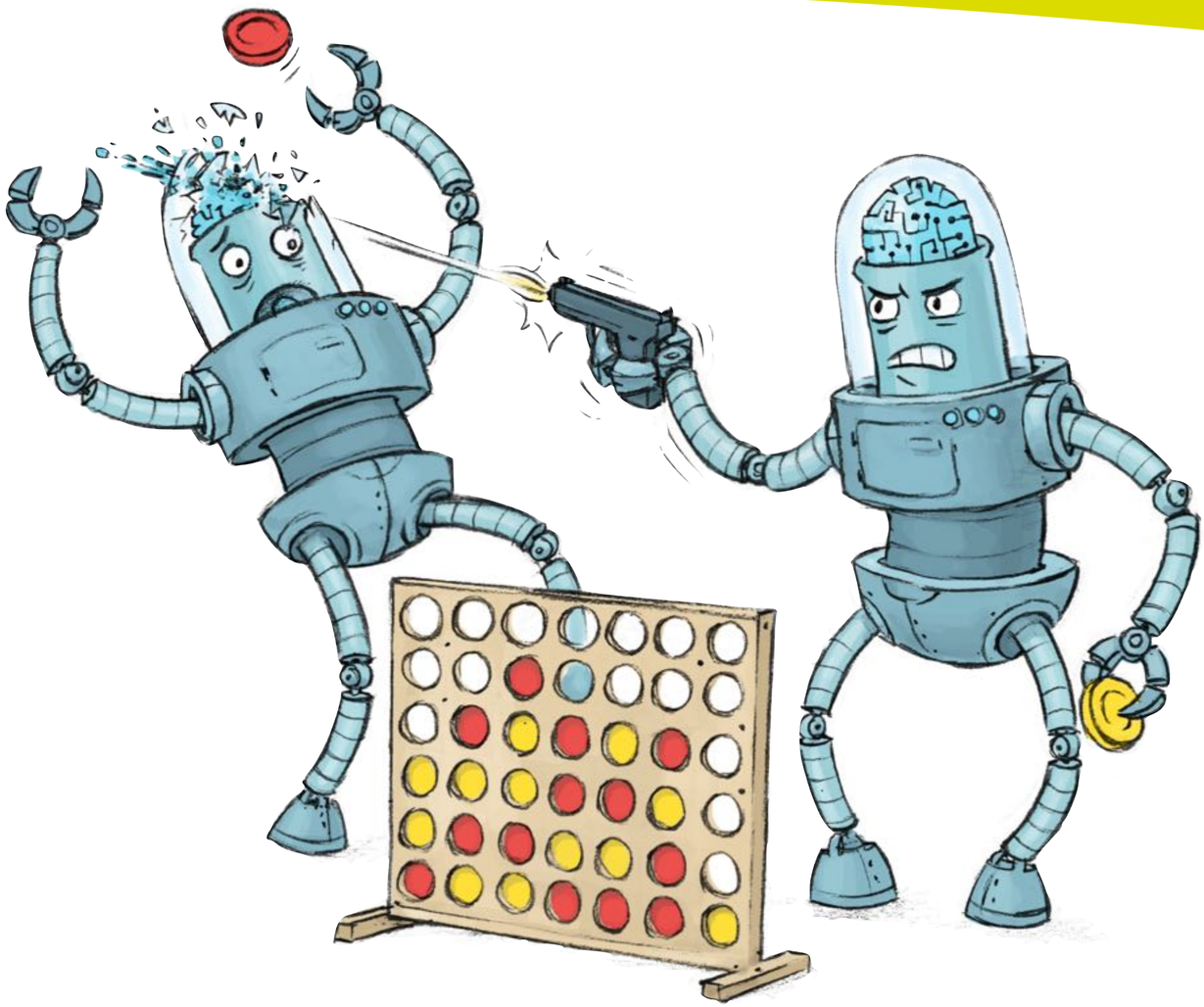
Agent 1: `(2,2)`

Agent 2: `(-2147483647,-2147483647)`

Agent 1: `(1,2)`

Agent 2: `(2147483647,2147483647)`

Agent 1: `OutOfMemoryError`



Singularity

Stop-Button Problem

- Juhu, unsere KI ist endlich intelligent

Stop-Button Problem

- Juhu, unsere KI ist endlich intelligent genug, um Tee zu besorgen…

Stop-Button Problem

- Juhu, unsere KI ist endlich intelligent genug, um Tee zu besorgen…

Beispiel: Rob Miles - Intro to Ai Safety (Youtube)

Alignment Problem

- Für das erreichen des Zieles sind weniger Variablen zu beobachten als die Umgebung hergibt
- Agent nutzt Umgebung aus um sein Ziel einfacher zu erreichen
- Agent findet Wege, die vorher (vielleicht) unbekannt waren

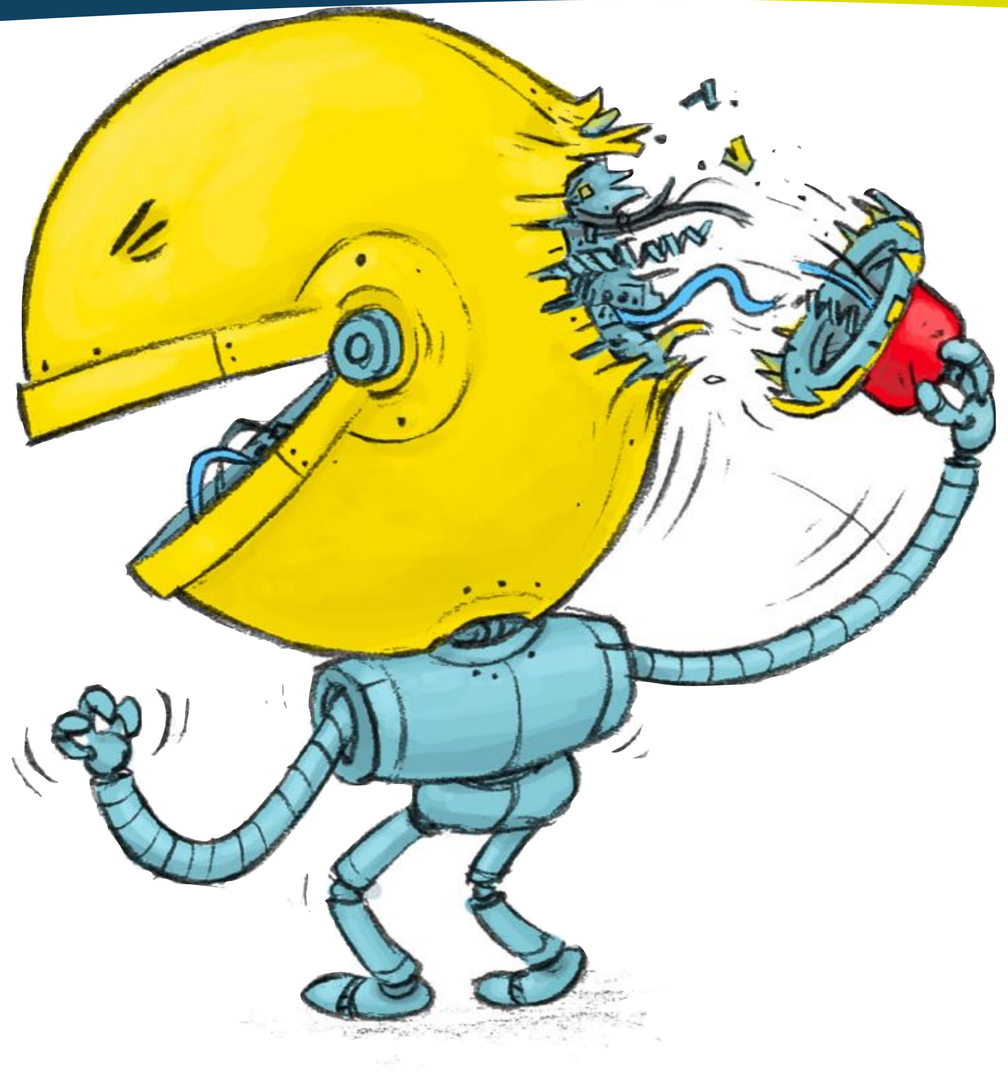
Stop-Button Problem

Stop-Button Problem

10

100

Stop-Button Problem

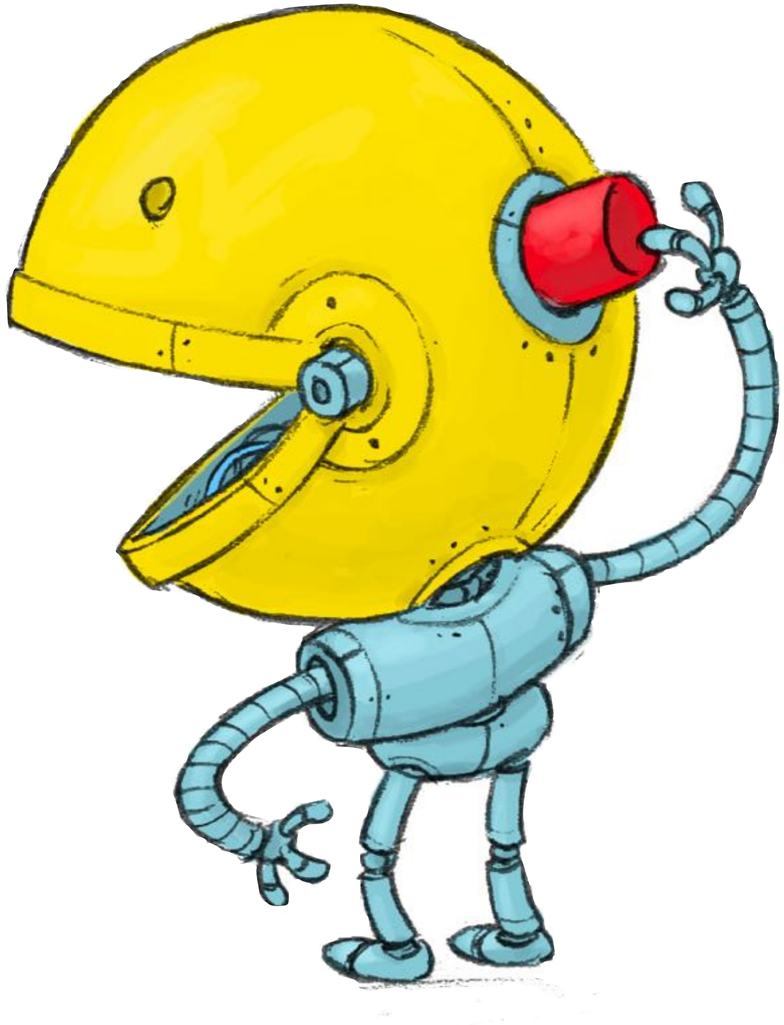


Stop-Button Problem

100

100

Stop-Button Problem



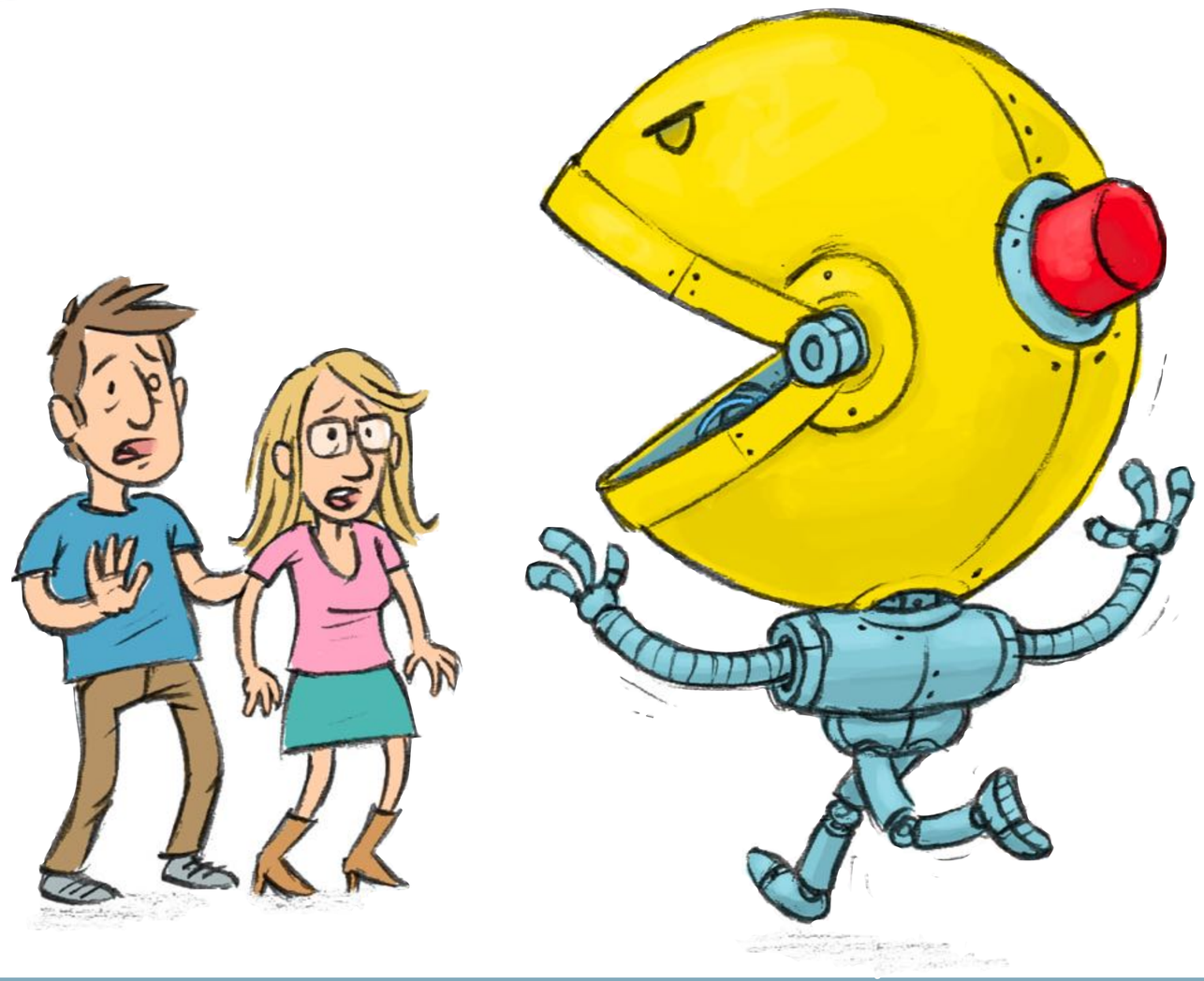
Stop-Button Problem

- Nur **ich** darf Button drücken

100

100

Stop-Button Problem



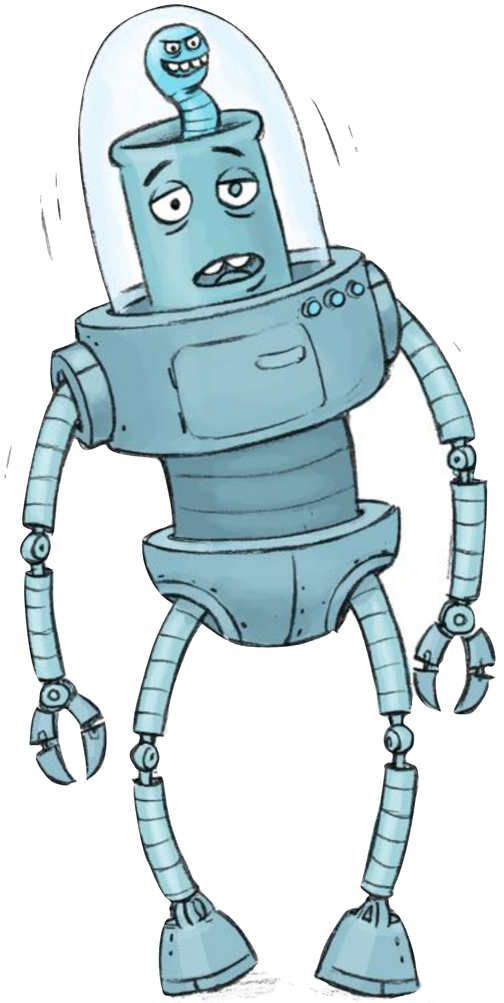
Mögliche Stop-Button Lösung?

- Einschränken der Sensoren des Agenten
- Ferngesteuerter „Stop-Button“
- Agent schaut Menschen zu und lernt richtiges Verhalten

Fazit & Ausblick

Fazit & Abmilderung (Ausblick)

- KI ist nicht böse
- Aktuelle KI dumm wie ein Wurm
- Zukunft: KI-Sicherheit muss das Rennen gegen super KI gewinnen



Weiteres zum Thema

- Rob (Robert) Miles - YouTube Kanal
- Janelle Shane - Künstliche Intelligenz (Buch)
- https://github.com/CKeibel/SingularityAusVersehenTalk

Digital Woche Dortmund (26.09. - 30.09.2022)

https://karakun.com/diwodo22/

Karakun DevHub_

dev.karakun.com

@C_Keibel

dev.karakun.com