

Version 1.0.4

Text: Marco Lardelli

© 2020-2021 Marco Lardelli.

Alle Urheber- und Leistungsschutzrechte sind vorbehalten. Dieses Buch ist nur für den privaten Gebrauch oder für den Gebrauch durch öffentliche Schulen (mind. zu 50% durch den Staat finanziert) bestimmt. Für alle Verwendungen, insbesondere Vorführung ausserhalb öffentlicher Schulen, Sendung, Bearbeitung und Vervielfältigung/Verbreitung bedarf es einer speziellen Bewilligung.

Inhalt Anhang B - Lösungen zu den Übungen

1. Was ist Intelligenz?	5
2. Intelligenz in der Natur: Nervensystem und Gehirn	8
3. Künstliche Intelligenz	9
4. Die Geschichte der künstlichen Intelligenz	10
5. Reinforcement Learning	10
6. Künstliche neuronale Netze	13
7. Algorithmen der künstlichen Intelligenz	17
8. Grundlagen der Robotik	19
9. Wir trainieren "Motti" in einer virtuellen 3D-Umgebung	20
10. Lokale Optima und Reward-Funktion	21
11. KI in unserem Alltag	21
12. KI-Ethik und KI-Sicherheit	22
13. Die Zukunft der KI	23
14. Künstliche Intelligenz und Philosophie	23

Anhang B Lösungen zu den Übungen

1. Was ist Intelligenz?

1.1 Definitionen

LÖSUNGEN ÜBUNG 1.1-1:

	Α	В	С	D	E	F
Smartphone	(X) 1	X 2	Х3	-	X 4	-
Mensch	(X) 1	X	X	X	X	X
Taschenrechner	(X) 1	X 5	Х3	-	-	-
Schachcomputer	(X) 1	X	X	-	-	-
Lernfähiger Roboter- Staubsauger	(X) 1	X 6	Х	(X) 7	-	(X) 8
Hund	(X) 1	(X)	X	X	X	X

Verschiedene Experten würden zu dieser Aufgabe vermutlich unterschiedliche Antworten geben. Es ist daher auch kein Problem, wenn deine Lösung von der obigen Tabelle abweicht. Wichtig ist, dass du dir dazu Gedanken gemacht hast.

- 1. Definition A ist insofern speziell, als dass sie die Definition auf den Intelligenztest verlagert. Von diesem hängt es natürlich ab, ob das System als intelligent gilt oder nicht.
- 2. Manche Apps verfügen über solche Fähigkeiten (z. B. Übersetzungs-App mit künstlicher Intelligenz, Siri) aber z. B. auch der Taschenrechner (siehe unten).
- 3. Taschenrechner: Die Wurzel von 2 zu ziehen ist (für Menschen!) ein schwieriges Problem. Für eine Maschine jedoch nicht.
- 4. Auf einem Smartphone findet man eine Vielzahl von Apps zur Lösung diverser Probleme.
- 5. Rechnen ist eine Form abstrakten Denkens.
- 6. Staubsauger-Roboter verfügen über raffinierte Algorithmen und oft auch KI, u. a. zur Planung und Steuerung des Reinigungsvorganges.
- 7. Staubsauger-Roboter funktionieren dank intelligenter Steuerung auf unterschiedlichem Untergrund (Teppich, Parkett) und mit unterschiedlichen Grundrissen.
- 8. Staubsauger-Roboter lernen mit der Zeit, besser Staub zu saugen. Ganz neue Fähigkeiten können sie hingegen nicht erwerben.

LÖSUNGEN ÜBUNG 1.1.-2:

Es fällt auf, dass die älteren Definitionen von heutigen technischen Systemen weitgehend erfüllt sind, die jüngeren jedoch nur teilweise.

Begründung:

Früher hat man Fähigkeiten als intelligent betrachtet, welche für Menschen schwierig auszuführen sind und mühsam erlernt werden müssen (Rechnen, Schach spielen). Es hat sich dann gezeigt, dass solche Fähigkeiten für Computer oft einfach zu bewältigen sind. Andere Fähigkeiten, die für uns (und z. T. sogar Tiere) selbstverständlich sind, z. B. die visuelle Wahrnehmung, Sprache oder die Motorik, waren jedoch mit künstlichen Systemen lange nur sehr schwer zu realisieren.

Nachdem dann auch auf diesen Gebieten Erfolge zu verzeichnen waren, hat man den Intelligenzbegriff um die allgemeine Fähigkeit, fast beliebige Fertigkeiten effizient zu erwerben, erweitert.

Auch die Abgrenzung vom Tier ("rationales Denken") war früher wohl wichtig, ist dies doch die einzige Eigenschaft (neben der Sprache), in der der Mensch dem Tier massiv überlegen ist.

1.2 Arten von Intelligenz

LÖSUNGEN ÜBUNG 1.2-1:

- Rationale Intelligenz: z. B. Fähigkeit zum logischen Denken (Mathematik betreiben, Schach spielen, Probleme lösen etc.).
- Emotionale Intelligenz: Fähigkeit, eigene und fremde Gefühle (korrekt) wahrzunehmen, zu verstehen und zu beeinflussen.
- Soziale Intelligenz: Fähigkeit, in Kommunikations- und Interaktionssituationen entsprechend den Bedürfnissen der Beteiligten effektiv zu handeln.
- Motorische Intelligenz: Fähigkeit, Bewegungen des Körpers effektiv zu steuern.

1.3 Verwandte Begriffe

LÖSUNGEN ÜBUNG 1.2-2:

Genie:

Seltene aussergewöhnlich ausgeprägte Intelligenz, oft nur auf einem engen Gebiet.

Talent, Begabung:

Potenzial, eine Fähigkeit sehr gut zu beherrschen. Muss nicht unbedingt entwickelt sein.

Kreativität:

Fähigkeit, überraschende Lösungen zu entwickeln. Intelligenz ist wohl eine Voraussetzung für Kreativität (bzw. Kreativität eine Form der Intelligenz), andererseits können intelligente Personen durchaus unkreativ sein.

Bsp.: Ein faszinierendes Bild zeichnen.

Wissen:

Gespeicherte Fakten.

Bsp.: 5 römische Kaiser aufzählen können.

Bildung:

Erworbene Fähigkeiten und erworbenes Wissen. Intelligenz vereinfacht die Bildung eines Menschen.

Bsp.: Eine Fremdsprache sprechen.

2. Intelligenz in der Natur: Nervensystem und Gehirn

2.1. Das Nervensystem und seine Unterteilungen

LÖSUNG ÜBUNG 2.1-1:

	willkürlich	vegetativ
Atmung		x
Herzschlag		х
Greifen mit der Hand	x	
Gehen	x	
Regelung des Blutdruckes		х

Z. T. sind vegetative Funktionen auch willentlich steuerbar. So kann man z. B. mit Yoga den Herzschlag beeinflussen oder man kann (für eine kurze Zeit) den Atem anhalten.

LÖSUNG ÜBUNG 2.1-2:

- Sehsinn
- Hörsinn
- Gleichgewichtssinn (z. B. wo ist oben und unten?)
- Geruchssinn
- Geschmackssinn (scharf, sauer, salzig etc.)
- Tastsinn (Berührung, Kälte, Hitze, Schmerz, Kraft, Lage, Bewegung)

LÖSUNG ÜBUNG 2.1-3:

A. Nematode (ein primitiver Wurm): 300

B. Fruchtfliege: 250'000

C. Meerschweinchen: 240'000'000. (240 Mio)

D. Katze: 760'000'000 (760 Mio)

E. Bonobo (Menschenaffe): 28'000'000'000 (28 Mia)

3. Künstliche Intelligenz

3.1 Begriffe

LÖSUNG ÜBUNG 3.1:

- A) Schachcomputer (1985):Künstliche Intelligenz (intelligent, aber nicht lernfähig).
- B) Automatische Erkennung von Personen auf Facebook (2020): Künstliche neuronale Netze mit bis zu 100 Millionen Neuronen: Deep Learning.
- C) Diktiersoftware (Sprache zu Text konvertieren) (2000): Lernfähig, aber noch kein Deep Learning (erst ab ca. 2013): Machine Learning.
- D) Diktiersoftware (Sprache zu Text konvertieren) (2020): Ab ca. 2013 mittels Deep Learning gelöst: Deep Learning.
- E) Smartphone App, welche einem Selfie einen Schnauz hinzufügt (2019): 2019 sehr wahrscheinlich mittels Deep Learning gelöst.

3.2 Arten von Machine Learning

LÖSUNG ÜBUNG 3.2:

- A. Supervised Learning.

 Typisches Klassifikationsproblem.
- B. Unsupervised Learning (sog. "Clustering"). Clustering wird bei Kundendaten oft angewendet, um Entscheidungsgrundlagen für das Marketing zu schaffen.
- C. Reinforcement Learning. Der Gewinn wird als Reward verwendet. Sog. "algorithmisches Trading".
- D. Reinforcement Learning. Die Distanz wird als Reward verwendet. Roboter werden so oft in virtuellen 3D-Umgebungen trainiert (billiger und gefahrlos).
- E. Supervised Learning.
 Die Software lernt aus bisherigen Käufen (z. B. welche weiteren Produkte haben frühere Kunden gekauft, welche die Produkte des aktuellen Kunden ebenfalls gekauft haben).

4. Die Geschichte der künstlichen Intelligenz

ÜBUNG 4.3-1:

- A. https://de.wikipedia.org/wiki/Turing-Test
- B. Der Test prüft, ob die Maschine über **menschenähnliche** Intelligenz verfügt. Diese beinhaltet auch die Eigenschaft, Fehler zu machen (d. h. sich gelegentlich unintelligent zu verhalten). Eine Superintelligenz würde im Test wohl schnell erkannt werden (z. B. wenn sie innert Sekundenbruchteilen komplizierte mathematische Probleme lösen kann).
- C. Die Rollen zwischen Mensch und Maschine sind vertauscht.

ÜBUNG 4.3-2:

- A. Es werden oft "seltsame", nicht ganz passende Antworten gegeben.
- B. ELIZA vermeidet es, wirklich Fragen zu beantworten (das wäre zu schwierig), sondern reagiert mit einer Gegenfrage, in welche es ein Stichwort der vorherigen Eingabe des Users einbaut.
 - Findet ELIZA kein Stichwort, das für eine Antwort verwendet werden kann, reagiert es mit "I see" oder "Please go on". Diese Antworten passen auf jede vorherige Eingabe des Benutzers und gaukeln Verständnis vor.

5. Reinforcement Learning

5.1 Problemstellung und Begriffe im RL

LÖSUNG ÜBUNG 5.1:

- A. Der RL-Agent versucht den Return (die Summer der Rewards) zu maximieren.
- B. Im RL wird nach einer Policy gesucht (d. h. einer Verhaltensweise des Agenten), welche einen möglichst hohen Return erzeugt.
- C. Ein negativer Reward bedeutet eine Bestrafung des Agenten.

5.2 Beispiel "multi armed bandit"

ÜBUNG 5.2:

- A. Der Spieler.
- B. Die Geldbeträge, welche die Automaten anzeigen.
- C. Die Summe aller Geldbeträge eines Spieles.
- D. Alles, was im Inneren des Computers passiert (Programm, Speicher etc.).
- E. Nichts (dies ist ein Spezialfall)! Wir sehen nicht, wie der Computer im Inneren die Zufallszahlen erzeugt.
- F. 100 Zeitschritte.

5.3 Der "exploitation / exploration trade-off"

ÜBUNG 5.3-1:

Das Problem verändert sich im Laufe des Spiels ("nicht stationär"). Deshalb muss man fortlaufend nach dem besten Automaten suchen und nicht nur am Anfang des Spiels.

ÜBUNG 5.3-2:

Epsilon-Greedy ist erfolgreicher. Der Grund ist, dass das Problem sich im Laufe der Zeit verändert. Es ist also wichtig, dass immer wieder geprüft wird, welcher Automat z.Z. gerade der Beste ist.

5.4 "Motti" der lichtsuchende Roboter

ÜBUNG 5.4-1:

Wird der Roboter mit der Taschenlampe teilweise beleuchtet, versucht er in den Lampenkegel zu fahren. Ist die Taschenlampe jedoch weit weg vom Roboter, kann er das Licht nicht finden. Hat er das Licht einmal verloren, so findet er meistens nicht mehr in den Lichtkegel zurück.

ÜBUNG 5.4-2:

Es wäre möglich, aber ein gutes Resultat zu erzielen, ist nicht ganz einfach:

- Es gibt sehr viele Kombinationsmöglichkeiten von teilweise beleuchteten Sensoren, die berücksichtigt werden müssen (z. B. "vorne sehr hell und nur rechts etwas hell").
- Das Programm muss bei starkem wie auch schwachem Licht funktionieren (verschiedene Taschenlampen).
- Das Programm muss bei unterschiedlich hellem Umgebungslicht funktionieren.

ÜBUNG 5.4-3:

- A. Die Software, die auf dem Roboter läuft (die Hardware des Roboters gehört zur Umgebung!).
- B. Die Summe des gemessenen Lichteinfalls auf die vier Lichtsensoren.
- C. Der Summe von B über die Laufzeit des Roboters.
- D. Die Roboter-Hardware und alles darum herum (Boden, Taschenlampe etc.).
- E. Der Lichteinfall an vier Punkten.
- F. Solange der Roboter eingeschaltet ist.

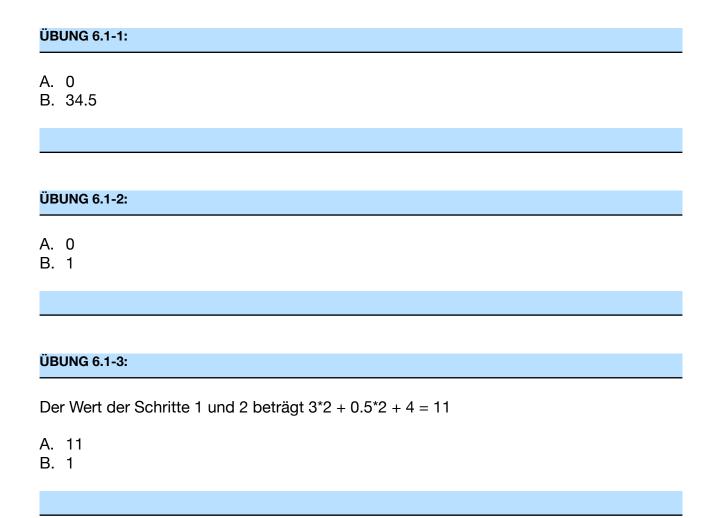
ÜBUNG 5.4-4 (SCHWER):

Mögliche Verbesserungen wären u. a.:

- Bessere Eichung der Sensoren (z. B. in zwei Zuständen dunkel und ganz hell).
- Mehr Beleuchtungssituationen unterscheiden.

6. Künstliche neuronale Netze

6.1 Künstliche Neuronen



ÜBUNG 6.1-4:

Ohne Aktivierungsfunktion wäre der Output des Neurons:

Output =
$$W_1 * I_1 + W_2 * I_2 + B$$

Genau am Punkt an welchem die Step-Funktion von 0 auf 1 springt (dort ist die Grenze der Flächen), muss der Output gleich 0 sein. Wir setzen also die obige Gleichung gleich Null:

$$0 = W_1 * I_1 + W_2 * I_2 + B$$

Nun können wir diese Gleichung so umformen, dass I₁ alleine auf einer Seite steht:

$$I_1 = (-W_2/W_1) * I_2 - B/W_1$$

Dies ist nun die Gleichung einer Geraden mit Steigung (- W_2/W_1) und y-Achsenabstand (- B/W_1).

6.2 Eigenschaften künstlicher Neuronen

ÜBUNG 6.2:

- A. Ja (ausser für den extrem seltenen Fall, dass die Punkte genau zusammenfallen).
- B. Ja (ausser u.U. wenn Punkte unterschiedlicher Farbe zusammenfallen).
- C. Nein
- D. Nein

Wir sehen, dass einzelne Neuronen nur sehr einfache Probleme lösen können.

6.3 Künstliche neuronale Netze (KNN)

ÜBUNG 6.3-1:

Die Motoren können vorwärts und rückwärts laufen, und haben eine variable, aber begrenzte Kraft. Ideal ist daher eine Aktivierungsfunktion, die einen beschränkten Wertebereich hat. Es eignet sich z. B. der Tangens Hyperbolicus.

ReLU ist nicht beschränkt (d. h. der Output kann beliebig gross werden) und zudem könnte der Roboter nicht rückwärtsfahren. Sie eignet sich daher nicht.

Die Step-Funktion könnte man durch Subtraktion von 0.5 auch dazu bringen, vorwärts und rückwärts zu unterscheiden, dann jedoch jeweils nur mit voller Kraft ohne Zwischenstufen.

Was auch möglich ist: keine Aktivierungsfunktion. Dann muss man aber die Motorkraft auf erlaubte Werte begrenzen. In den Roboter-Programmen im Kit ist diese Lösung gewählt.

ÜBUNG 6.3-2:

Wichtig ist es, zu verstehen, dass neuronale Netze letztlich auf Geometrie basieren (und nicht z. B. auf Sprache oder Logik).

ÜBUNG 6.3-3:

Siehe das zusätzliche Material (PDFs) im Download-Bereich der Website. Es ist für jedes Neuron ein PDF vorhanden (eines für die Schüler und eines ausgefüllt als Lösung).

- A. Wenn nur Licht auf den vorderen Sensor fällt, sind die Outputs beider Output-Neuronen positiv. D. h. der Roboter fährt vorwärts. Dies ist korrekt.
- B. Wenn nur Licht auf den hinteren Sensor fällt, sind die Outputs beider Output-Neuronen negativ. D. h. der Roboter fährt rückwärts. Dies ist korrekt.

ÜBUNG 6.3-4 (SCHWER):

Der Output des Neurons lautet:

Bezeichnungen (Beispiele):

1 und 2 sind die beiden "hidden layer"-Neuronen, 3 ist das Output-Neuron. W1₂ ist das zweite Gewicht des ersten Neurons. B3 ist der Bias des dritten Neurons.

Output =
$$(I_1*W1_1 + I_2*W1_2 + B1)*W3_1 + (I_1*W2_1 + I_2*W2_2 + B2)*W3_2 + B3$$

$$= I_1^* (W1_1 + W2_1) + I_2^* (W1_2 + W2_2) + (B3 + B1^*W3_1 + B2^*W3_2)$$

Die letzte Gleichung entspricht derjenigen eines einzelnen Neurons mit Gewichten und Bias entsprechend den Termen in Klammern.

Dies ist der Grund, warum es unbedingt eine Aktivierungsfunktion braucht. Ohne sie machen komplexe neuronale Netze mit mehreren Schichten keinen Sinn, da ihre Komplexität auf diejenige weniger Neuronen beschränkt wäre.

Genau genommen darf die Aktivierungsfunktion auch **nicht linear** sein (d. h. sie darf im Plot keine Gerade darstellen, sondern muss irgendwo gekrümmt oder geknickt sein).

7. Algorithmen der künstlichen Intelligenz

7.1 Das Problem des grossen Suchraumes

ÜBUNG 7.1-1:

- 1. Jedes Neuron im "hidden layer" hat 5 Parameter (für jedes der vier Neuronen im "input layer" ein Gewicht plus ein Bias). Insgesamt also 50 Parameter in diesem Layer.
- Jedes der beiden Neuronen im "output layer" hat 11 Parameter (für jedes der 10 Neuronen im "hidden layer" ein Gewicht plus ein Bias). Insgesamt also 22 Parameter in diesem Layer.
- 3. Insgesamt hat "Motti" also 72 Parameter!

ÜBUNG 7.1-2:

5*5 = 25

ÜBUNG 7.1-3:

- A. 10
- B. 100
- C. 1000

ÜBUNG 7.1-4:

Natürlich kann man nur bis zu 3 Dimensionen auf natürliche Weise visualisieren. Ab "4D" werden zusätzliche Würfel einfach nebeneinander gestellt.

7.2 Der genetische Algorithmus

ÜBUNG 32:

Der Spieler übernimmt den Evaluationsprozess (ist die Kreatur dem Ziel ähnlich?) und den Selektionsprozess (Auswahl der besten Kreaturen).

ÜBUNG 7.2-2:

- A. Evolutionstheorie
- B. Es fehlt die Sexualität. Es wäre möglich (und u.U. auch vorteilhaft), nach der Selektion neue Kreaturen durch Rekombination (d.h. Mischen) der Parameter von zwei Kreaturen zu erzeugen.

7.3 Der "exploitation / exploration trade-off"

ÜBUNG 7.3:

- C. Mehr "exploration", weniger "exploitation" (der Algorithmus konvergiert - d.h. findet die Lösung - weniger schnell, findet aber wahrscheinlich eine bessere Lösung).
- D. Mehr "explo**it**ation", weniger "explo**r**ation" (der Algorithmus konvergiert schneller, das Resultat ist aber wahrscheinlich schlechter).

7.4 Die ideale Grösse der Generation. Hyperparameter

ÜBUNG 7.4:

- A. Es gibt 6 Parameter, d. h. die Generationengrösse sollte nach der Faustregel 60 betragen. Die tatsächliche Grösse beträgt aber nur 16. Daher zu klein.
- B. Für 72 KNN Parameter wäre eine Generationengrösse von ca. 720 nötig.

8. Grundlagen der Robotik

ÜBUNG 8.-1:

- Vier Lichtsensoren.
- · Zwei Motoren, welche Räder antreiben.
- Dem micro:bit und der Steuerelektronik für die Motoren.

ÜBUNG 8.-2:

- A. Alle Sinnesorgane
- B. Arme, Beine, Hände etc. mit den Muskeln als "Motoren"
- C. Das Zentralnervensystem

ÜBUNG 8.-3:

- 1. Humanoider Roboter
- 2. Drohne
- 3. Mars-Rover
- 4. Industrie-Roboter

9. Wir trainieren "Motti" in einer virtuellen 3D-Umgebung

ÜBUNG 9.-1:

Die Einstellung der richtigen FPS ist wichtig, damit keine Rechenleistung verschenkt wird.

ÜBUNG 9.-2:

Der Trainingsvorgang braucht etwas Zeit. Er kann z. B. während der Mittagspause durchgeführt werden. Der Computer sollte dazu so konfiguriert sein, dass er nicht nach ein paar Minuten in einen Schlafmodus fällt.

ÜBUNG 9.-3:

Eine Anleitung zum Programmieren der Roboter findet du auch im Anhang (das Buch mit den Bauanleitungen).

ÜBUNG 9.-4 (SCHWER):

- A. Sie befinden sich am Anfang des Codes.
- B. Am besten suchst du nach den Aktivierungsfunktionen (siehe B, C), dann findet du den KNN-Code recht einfach.
- C. if (Accum < 0) { Hidden[i] = 0.1 * Accum } else { Hidden[i] = Accum } (das ist Leaky ReLU!)
- D. Z. B. Output2[k] = Accum2 (falls keine Aktivierungsfunktion gewählt wurde).

10. Lokale Optima und Reward-Funktion

ÜBUNG 10.:

Falls mehrere schnelle Rechner vorhanden sind, kannst du (bzw. die Klasse/Gruppe) mit verschiedenen Werten für die Bestrafung experimentieren (z. B. 1000, 2000, 3000, 5000).

11. KI in unserem Alltag

12. KI-Ethik und KI-Sicherheit

ÜBUNG 12.3:

- A. 1% von 10'000 sind 100. 100 krank, 9'900 gesund.
- B. Bei den 100 Kranken irrt sich die KI in 10% der Fälle. Es gibt also 90 "true positives".
- C. Es gibt entsprechend 10 "false negatives".
- D. Bei den 9'900 Gesunden irrt sich die KI auch in 10% der Fälle. Es gibt also 990 "false positives".
- E. 990 "false positives" für nur 90 "true positives" sind vermutlich zu viel. Also eher **nicht** brauchbar.

13. Die Zukunft der KI

14. Künstliche Intelligenz und Philosophie

ÜBUNG 45:

https://de.wikipedia.org/wiki/Chinesisches Zimmer

1.,2. und 4. sind vermutlich zu erreichen. Bei diesem Gedankenexperiment geht es um 3. , also das Bewusstsein.