
Unterrichtseinheit zum Thema “Künstliche Intelligenz” (SEK II)

Willkommen zurück!

Marcel Hahn, Florian D’Uva
Finn Ohlrogge, Paul Schulze
finn.ohlrogge@uni-rostock.de

Universität Rostock - Institut für Informatik
16. November 2021

Gliederung

1. Begrüßung
2. Einrichtung des Systems und der Programmierumgebung
3. SuS-Aufgaben: Modellieren statt Programmieren
4. Ausprobieren, Testen und Rumspielen
5. Auswertung der Implementation (Woche 6)
6. *Ethik und gesellschaftliche Verantwortung (Woche 8)*

Einrichtung des Systems und der Programmierumgebung

Systemanforderungen und Hinweise

- Python 3.6-3.9
 - <https://www.python.org/downloads/release/python-397/>
 - Verzeichnis merken (Appdata)
- Softwareprojekt
 - <https://github.com/Schupax/KI-OCR-Schulprojekt>
- Installationsdauer 15-30 Minuten

Los geht's!

- individuelle Betreuung
 - Live-Installation im Breakout-room
-

SuS-Aufgaben:

Modellieren statt programmieren

Vorbereitung der Entwicklungsumgebung für die SuS

1. Download des Projektes

<https://github.com/Schupax/KI-OCR-Schulprojekt>

2. UML-Ordner entfernen/löschen
 3. Vorlagen und Aufgaben vorbereiten
 - a. Layer.py ersetzen
 - b. Konsolenanwendung.py ersetzen
 - c. kopieren/verschieben
 - i. erstelleLayerMitFehlern.nsd
 - ii. erstelleNetzwerk.nsd
-

Klasse & Validierung implementieren

Layer
-Neuronenanzahl: int -Layertyp: int
+Layer(pNeuronenanzahl:int,pLayertyp:int) +getLayertyp(): int +setLayertyp(pLayertyp:int): void +getNeuronenanzahl(): int +setNeuronenanzahl(pNeuronenanzahl:int): void

Klasse & Validierung implementieren

```
1 public class Layer {
2
3     private int layerType;
4     private int neuronAnzahl;
5
6     public Layer(int pLayerType, int pNeuronenAnzahl) {
7         setLayerType(pLayerType);
8         setNeuronAnzahl(pNeuronenAnzahl);
9     }
10
11     public int getLayerType() {
12         return layerType;
13     }
14
15     public void setLayerType(int pLayerType) {
16         if (pLayerType > 0 && pLayerType < 4) {
17             this.layerType = pLayerType;
18         } else {
19             System.out.println("Setzen des Layertyp ist fehlgeschlagen");
20         }
21     }
22
23     public int getNeuronAnzahl() {
24         return neuronAnzahl;
25     }
26
27     public void setNeuronAnzahl(int pNeuronAnzahl) {
28         if (pNeuronAnzahl > 0) {
29             this.neuronAnzahl = pNeuronAnzahl;
30         } else {
31             System.out.println("Setzen der Neuronenanzahl ist fehlgeschlagen");
32         }
33     }
34 }
```

```
1 """
2     Diese Modellklasse fasst die Eigenschaften und Funktionen/Methoden eines
3     einer Schicht in einem neuronalen Netz zusammen.
4 """
5
6 class Layer(object):
7
8     """
9     mögliche Werte für den "LayerType" und ihre Bedeutung
10     1 --> Inputlayer
11     2 --> Hiddenlayer
12     3 --> Outputlayer
13     """
14     def __init__(self, pNeuronAnzahl, pLayerType):
15         self.neuronAnzahl = pNeuronAnzahl
16         self.layerType = pLayerType
17
18     def gibNeuronenAnzahl(self):
19         return self.neuronAnzahl
20
21     def setzeNeuronenAnzahl(self, pNeuronenAnzahl):
22         if pNeuronenAnzahl > 0:
23             self.neuronAnzahl = pNeuronenAnzahl
24         else:
25             print("Setzen der Neuronenanzahl ist fehlgeschlagen")
26
27     def gibLayerType(self):
28         return self.layerType
29
30     def setzeLayerType(self, pLayerType):
31         if pLayerType > 0 and pLayerType < 4:
32             self.layerType = pLayerType
33         else:
34             print("Setzen des Layertyp ist fehlgeschlagen")
```

Klasse & Validierung implementieren

Java		Python
this	→	self
public [T]	→	def



Fehler erkennen

- In der Methode *erstelleLayer* sind 4 Fehler versteckt
 - 1 x Syntaxfehler
 - 2 x Logikfehler
 - 1 x fachlicher Fehler

erstelleLayer

neuronenanzahl ← -1

layertyp ← -1

neuronenanzahl == -1

print("Bitte gib die Anzahl der Neuronen für den Layer ein:")

eingabe ← input()

eingabe > 0

T

F

eingabe ← neuronenanzahl

print("die Eingabe muss größer als 0 sein!")

layertyp == -1

print("Bitte gib einen Layertyp an (Werte von 1-3):")

print("1 --> Inputlayer")

print("2 --> Outputlayer")

print("3 --> Hiddenlayer")

eingabe ← input()

eingabe > 0 && eingabe < 4

T

F

print("die Eingabe muss größer als 0 sein und kleiner als 4!")

layertyp ← eingabe

return Layer(neuronenanzahl, layertyp)

Fehler erkennen - Lösung

- In der Methode *erstelleLayer* sind 4 Fehler

- Die Zuweisung ist falsch herum (Logikfehler)
- Am Ende der Zeile fehlt eine Klammer) (Syntaxfehler)
- Vertauschen der Eingabe der Layertypen
- Die Sequenzen müssen getauscht werden

1)

2) & 3)

4)

erstelleLayer

```
neuronenanzahl ← -1
```

```
layertyp ← -1
```

```
neuronenanzahl == -1
```

```
print("Bitte gib die Anzahl der Neuronen für den Layer ein.")
```

```
eingabe ← input()
```

```
eingabe > 0
```

T

F

```
eingabe ← neuronenanzahl
```

```
print("die Eingabe muss größer als 0 sein")
```

```
layertyp == -1
```

```
print("Bitte gib einen Layertyp an (Werte von 1-3):")
```

```
print("1 --> Inputlayer")
```

```
print("2 --> Outputlayer")
```

```
print("3 --> Hiddenlayer")
```

```
eingabe ← input()
```

```
eingabe > 0 && eingabe < 4
```

T

F

```
print("die Eingabe muss größer als 0 sein und kleiner als 4")
```

```
layertyp ← eingabe
```

```
return Layer(neuronenanzahl, layertyp)
```

Algorithmus vervollständigen

- Vervollständigen Sie den Algorithmus für die Erstellung eines neuronalen Netzes!

erstelleNetzwerk

aktOption=-1

optionen = ["abbrechen", "Layer anlegen", "Netzwerkerstellung abschließen"]

layers = []

Algorithmus vervollständigen

- Eingabe &-aufforderung
- Schleife
- Verzweigungen
- Liste
- Objekte erzeugen
- Methode benutzen

erstelleNetzwerk

aktOption ← -1

optionen ← ["abbrechen", "Layer anlegen", "Netzwerkerstellung abschließen"]

layers ← []

while (**aktOption** != 0 and **aktOption** != 2)

print("Die folgenden Optionen stehen zur Verfügung:")
print("1 für abbrechen")
print("2 für Layer anlegen")
print("3 für Netzwerkerstellung abschließen")

eingabe = input()

(eingabe > 0 and eingabe < 4)

T

F

aktOption ← eingabe - 1

∅

(**aktOption** == 1)

T

F

layer ← erstelleLayer()
layers.append(layer)

∅

(**aktOption** == 2)

T

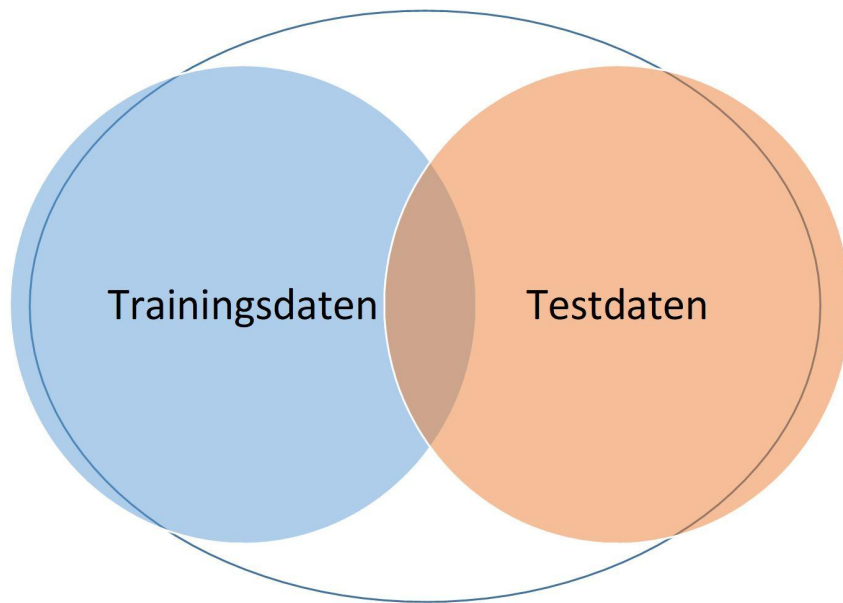
F

dataset ← Dataset()
dataset.loadMNIST()
self.netz = NeutralesNetz(**layers**, Regelwerk(), **dataset**)

∅

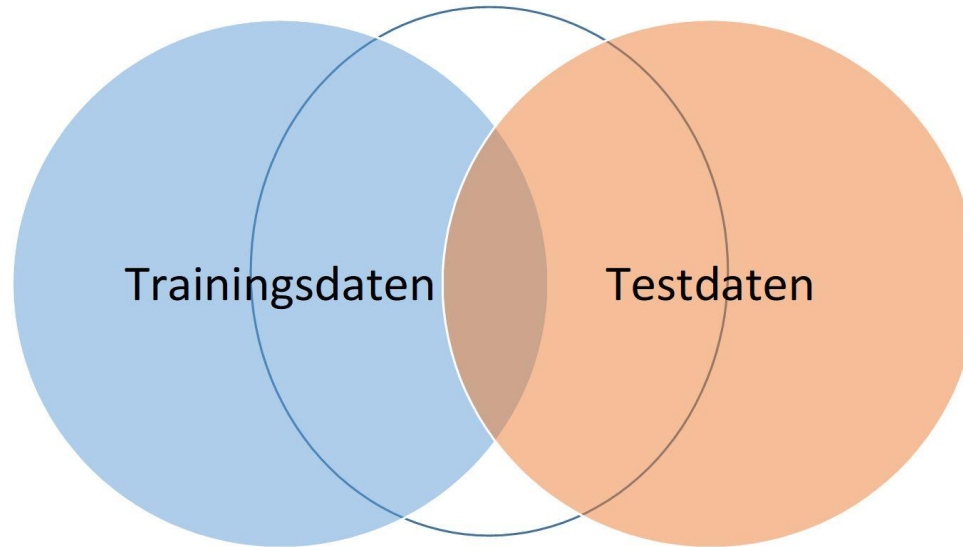
Ausprobieren & Testen & Rumspielen

Mögliche Szenarien der Ziffernerkennung



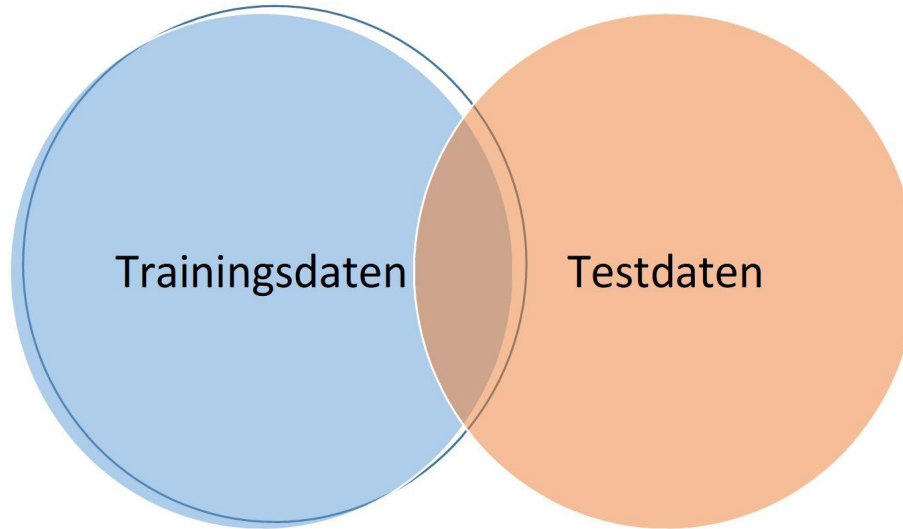
Optimum: wenige Trainingsfehler, geringe Testfehleranzahl

Mögliche Szenarien der Ziffernerkennung



Underfitting: viele Fehler sowohl bei Trainings- als auch Testdaten

Mögliche Szenarien der Ziffernerkennung



Overfitting: wenige Trainingsfehler, hohe Testfehleranzahl

Aufgabe: Erstellen von Netzwerken

Erstellt die verschiedenen neuronalen Netzwerke, trainiert die KI und überprüft die Genauigkeit der KI mithilfe eigener Eingaben.

Ohne Hidden-Layer:

- 1 Neuron Input-, 10 Output-Layer
- 10 Input, 10 Output
- 784 Input, 10 Output

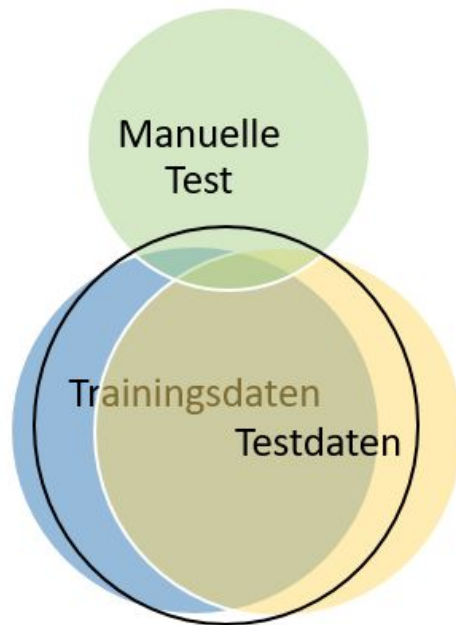
Mit Hidden-Layer:

- 10 Input, 10 Hidden, 10 Output
- 10 Input, 100 Hidden, 10 Output

Optimum: 784 Input, 200 Hidden, 100 Hidden, 10 Output?

Auswertung der Implementation (Woche 6)

1. KI mit wenigen Neuronen sehr gut
(10 Input, 10 Output -> über 90 % Genauigkeit)
2. Weniger Hidden als Input hemmt KI
(10 Input, 1 Hidden, 10 Output -> ca. 25 %)
3. Größere Anzahl an Neuronen verbessern die
Genauigkeit hinsichtlich Trainings- und Testdaten
4. Aber: Manuelle Eingabe in allen Fällen schlechter
Warum?



Es folgt ein Blick auf die Trainings- und Testdaten.

Welche Zahl wird gesucht?

- Einstieg -



Label = 8



Label = 6



Label = 9



Label = 4

Welche Zahl wird gesucht?

- Für Fortgeschrittene -



Label = 3



Label = 1



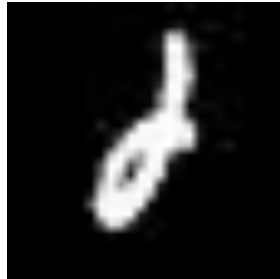
Label = 7



Label = 0

Welche Zahl wird gesucht?

- Für Fortgeschrittene -



Label = 2



Label = 4



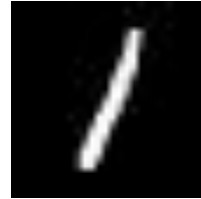
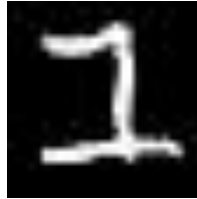
Label = 2



Label = 8

Unterschiede in der Schreibweise

Nummer 1:



Nummer 4:



Nummer 7:



Bis hierher Fragen?

Ethik und gesellschaftliche Verantwortung

Ethik und gesellschaftliche Verantwortung (Woche 8)

Ziele:

Die Schülerinnen und Schüler sind nach der Unterrichtseinheit in der Lage:

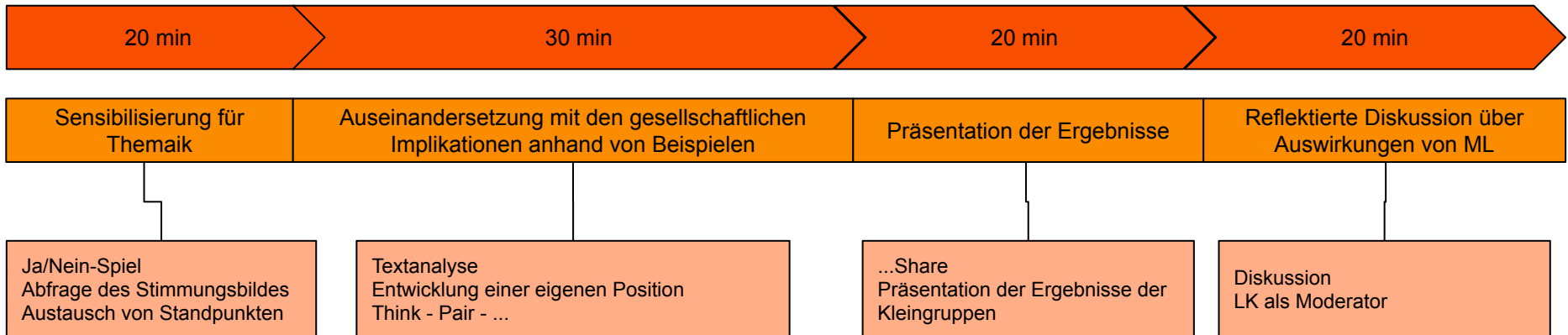
- den Einsatz des Informatiksystems KI unter fachlichen und darüber hinaus unter ethischen, sozialen, ökologischen, medizinischen, ökonomischen oder rechtlichen Aspekten bewerten
- entwickeln im gemeinsamen Dialog ein Bild, wie Technik gestaltet sein sollte, um dem Menschen dienlich zu sein und ein gutes gesellschaftliches Zusammenleben zu fördern

Motivation:

- Wie ethisch ist KI? - Science Slam (<https://www.youtube.com/watch?v=Pzx2wkba6tc>)
-

Ethik und gesellschaftliche Verantwortung

- zeitlicher Ablauf -



Ja/Nein-Spiel

Material:

- “Ja” & “Nein”-Karten (grün/rot) für die S&S
- vorbereitete Frage (Checkpoll: <https://www.checkpoll.de/ 5ub4824qw.html>)

Fragen: (variabel, je nach schon angesprochenen Thematiken)

1. Sollten Pakete in Zukunft von autonomen Fahrzeugen ausgeliefert werden?
 2. Sollten Menschen und Roboter in Zukunft kooperativ zusammenarbeiten, z.B. im Pflegebereich?
 3. Brauchen wir Roboter?
 4. Sollte eine autonome Kriegsführung möglich sein?
-

Beispielthemen Machine Learning

/i/ Autonomes Fahren

Schon längst ist autonomes Fahren viel mehr als nur ein Forschungsthema. Verschiedene Unternehmen haben in den letzten Jahren überzeugende Entwicklungen gezeigt, die den Einsatz in immer greifbarere Nähe rücken lassen.

Im Prinzip bewegen sich autonome Fahrzeuge mithilfe von Kameras, Sensoren und Radargeräten, indem sie ihre gegenwärtige Position und ihr Ziel berechnen und ständig Abstände, Signale und Hindernisse in ihrer Verkehrsumgebung messen bzw. erfassen und darauf in ihrer Navigation reagieren.

Gegenwärtig wird überwiegend an der Entwicklung von autonom fahrenden Shuttlebussen gearbeitet, die mehrere Personen transportieren können. Im Unterschied zu herkömmlichen Bussen und Bahnen, in denen der Fahrgast auch keine Fahraufgabe übernimmt, sollen diese Fahrzeuge individuelle Routen auf Abruf befahren können und haben keine/n menschlichen Fahrer*innen. Während der Fahrzeit, kann der Fahrgast die Zeit also frei nutzen, um zu arbeiten oder zu entspannen und ist darüber hinaus weder an Fahrpläne noch an feste Strecken gebunden.

Beim autonomen Fahren können einige Unfallsituationen und Sicherheitsrisiken für den Verkehr mit menschlichen Fahrer*innen entfallen, wie beispielsweise Alkohol am Steuer, irrationale Handlungen, Konzentrationsschwächen, Müdigkeit oder auch plötzlich auftretende gesundheitliche Probleme. Dennoch werden gefährliche Verkehrssituationen auftreten, auf die auch autonome Fahrsysteme reagieren müssen und die eine entsprechende Programmierung benötigen.

In Deutschland hat die Ethikkommission des BMVI (Bundesministerium für Verkehr und digitale Infrastruktur) dazu die weltweit ersten Leitlinien für autonomes Fahren formuliert. Darin werden Fragen des Datenschutzes, mögliche Gefahrensituationen und ihre Bewertung ebenso beschrieben wie Haftungsfragen bei Unfällen. Zu den Kernaussagen gehört u.a.: „Sachschaden geht vor Personenschaden: In Gefahrensituationen hat der Schutz menschlichen Lebens immer höchste Priorität. Bei unausweichlichen Unfallsituationen ist jede Qualifizierung von Menschen nach persönlichen Merkmalen (Alter, Geschlecht, körperliche oder geistige Konstitution) unzulässig.“ Und „in jeder Fahrsituation muss klar geregelt und erkennbar sein, wer für die Fahraufgabe zuständig ist: Der Mensch oder der Computer.“⁷

Weiterführende Links:

- Andreas Pfeffer (2018): Prognosen durch künstliche Intelligenz: Toyota verkürzt Wartezeit auf Taxis in Japan. www.elektroniknet.de/elektronik-automotive/assistenzsysteme/toyota-verkuerzt-wartezeit-auf-taxis-in-japan-151724.html
- Stefan Menzel (2017): Warum uns Robo-Autos neue Freiheiten schenken. edison.handelsblatt.com/ertraeumen/warum-uns-robo-autos-neue-freiheiten-schenken/20509078.html

⁷ Pressemitteilung des BMVI (08/2017): Ethik-Kommission zum automatisierten Fahren legt Bericht vor. www.bmvi.de/SharedDocs/DE/Pressemitteilungen/2017/084-dobrindt-bericht-der-ethik-kommission.html

Auswirkungen von ML auf die Gesellschaft

Welche Auswirkung hat Machine Learning auf Teile unserer Gesellschaft?

Wähle ein Thema und erarbeite gemeinsam zu zweit oder auch allein die Vorteile und Herausforderungen. Überlege zudem mögliche Richtlinien zur Vermeidung von Gefahren. Nutze hierfür die entsprechenden Textvorlagen.

Name/Titel

Bild. Du kannst auch eine Skizze anfertigen.

Kurze Beschreibung

Vorteile für die Gesellschaft

Herausforderungen und Gefahren für die Gesellschaft

Diskussion

Gedankenansätze zur abschließenden Diskussion

- Welchen Einfluss haben die Themen KI und ML auf das Zusammenleben in der Gesellschaft?
 - Wie könnte sich eine Gesellschaft durch ein Anwachsen an Produkten und Dienstleistungen verändern?
 - Wie kann man damit umgehen?
-