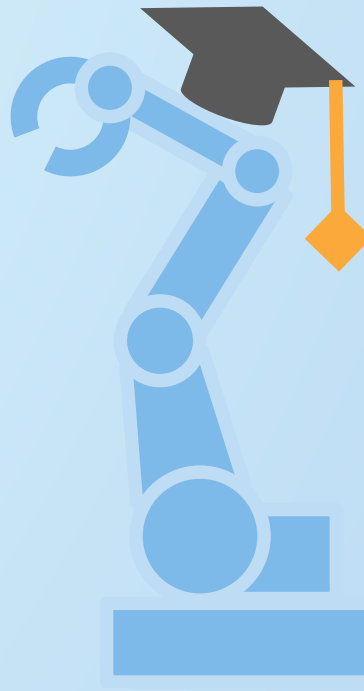




MACHINE LEARNING

EIN NEURONALES NETZ AUF DEM RASPBERRY PI 3

Sebastian Folz, Pi And More 11, Trier



ZUR PERSON

Sebastian Folz

Blog: <http://nobytes.blogspot.com>

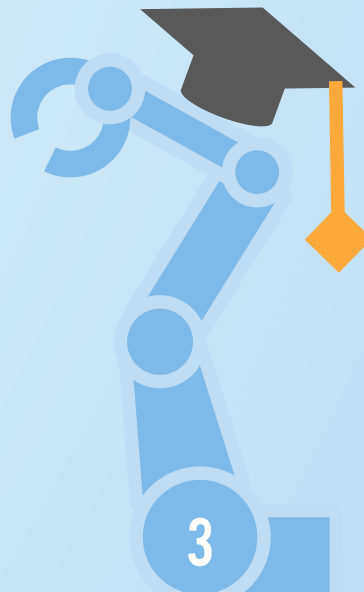
Twitter: [@FolzSebastian](https://twitter.com/FolzSebastian)

- Wirtschaftsinformatik, M.Sc., Hochschule Trier
- **Machine Learning Engineer**
- Datenanalyst / Data Scientist
- Software-Entwickler



AGENDA

1. Was ist ein neuronale Netz?
2. Problemstellung
3. Hardware und Software
4. Daten sammeln, normalisieren, splitten
5. Praxis und Resultate
6. Ungelöste Probleme



1. WAS IST EIN NEURONALES NETZ?

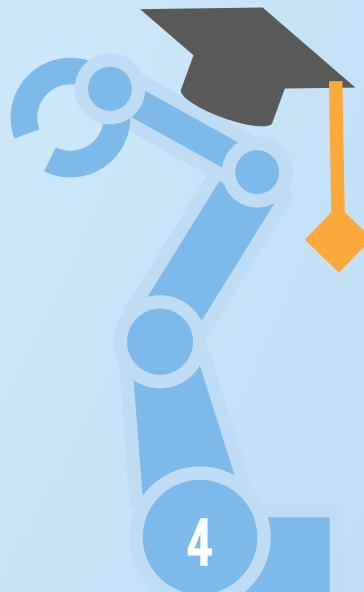
Definition künstliches Neuronales Netz:

»Künstliche Neuronale Netze (KNN) sind inspiriert durch *Nervenzellen* und lassen sich für maschinelles Lernen (Machine Learning) einsetzen. Mithilfe der KNN können verschiedene Problemstellungen computerbasiert gelöst werden.«

- Methode, um einem Computer *Wissen* zu vermitteln

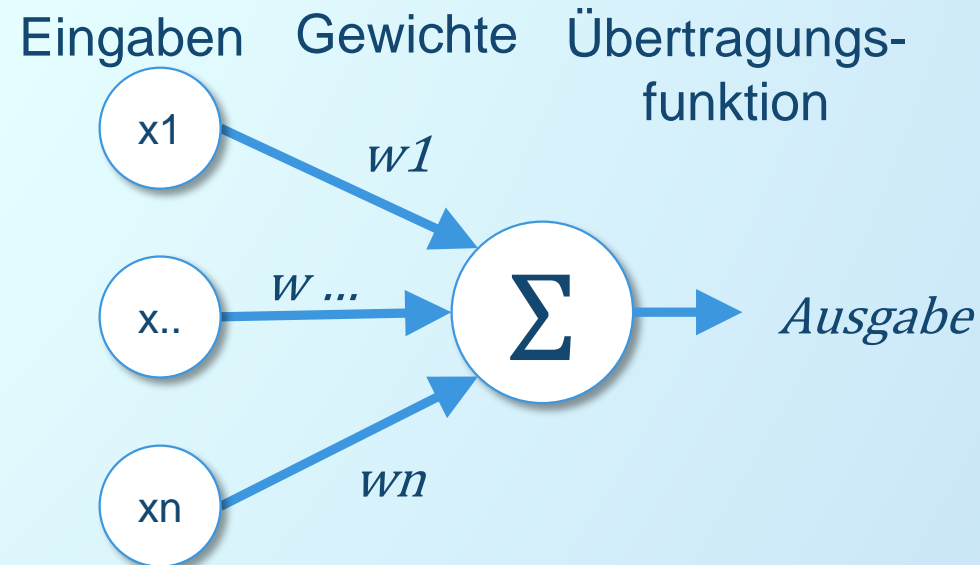
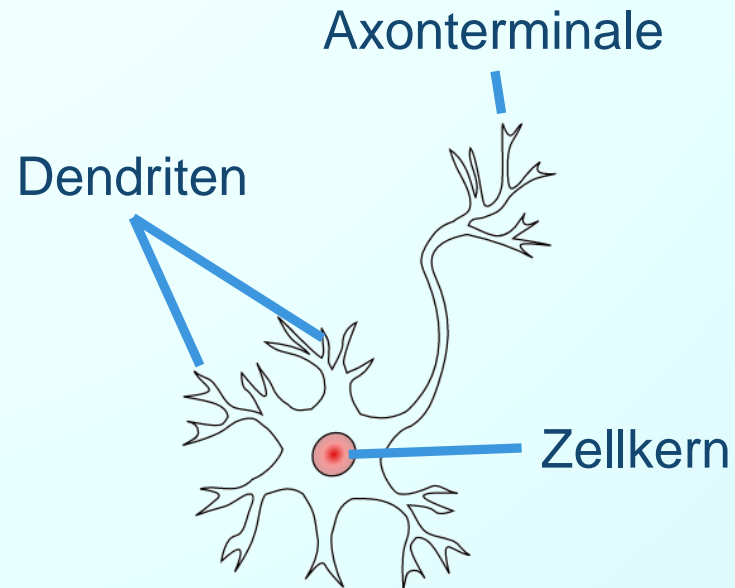
Angepasst übernommen von:

<https://www.bigdata-insider.de/was-ist-ein-neuronales-netz-a-686185/>



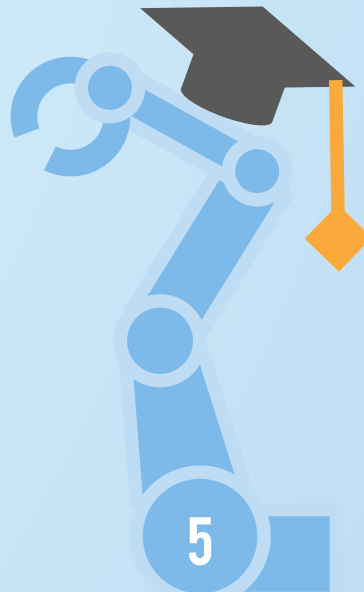
1. WAS IST EIN NEURONALES NETZ?

- Ein neuronales Netz (KNN) besteht aus künstlichen Neuronen
- Ein Neuron ist einer Nervenzelle nachgebaut



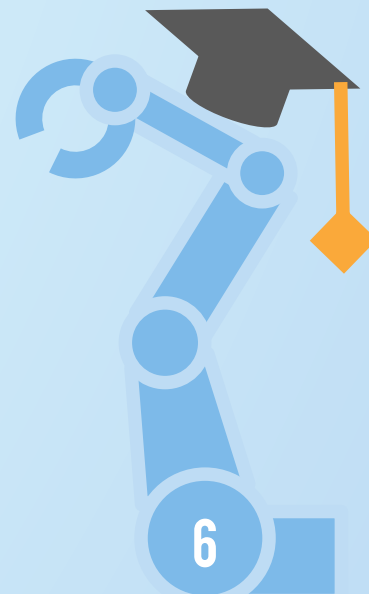
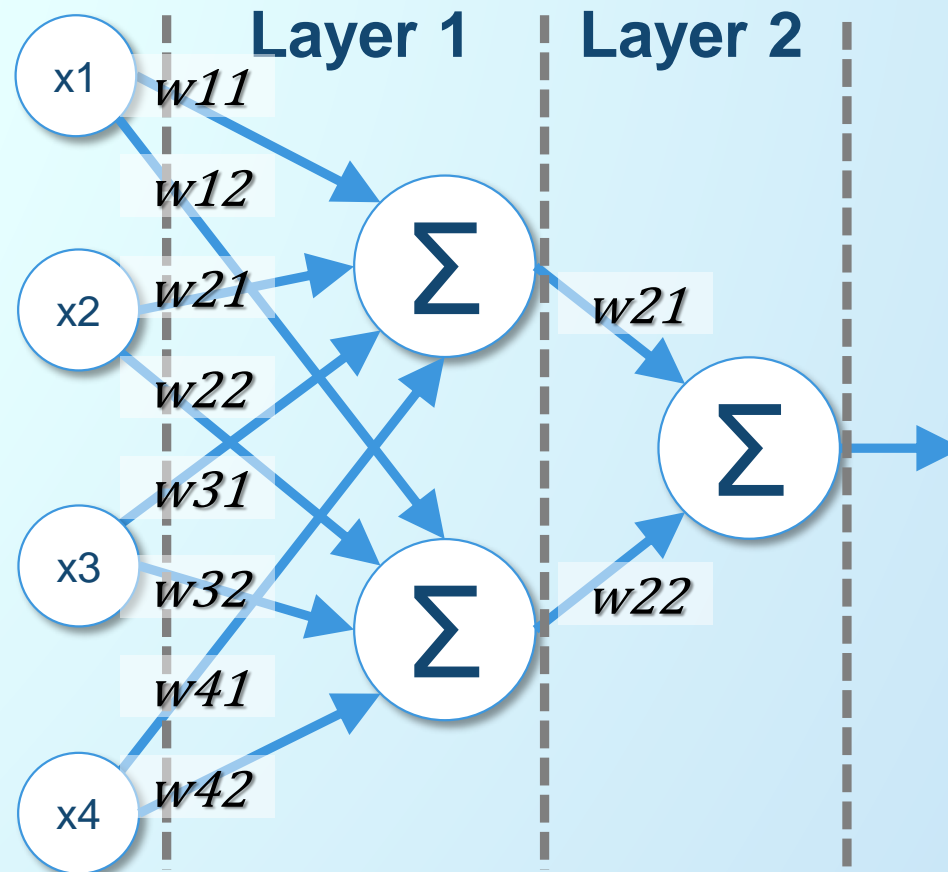
Bildquellen:

https://upload.wikimedia.org/wikipedia/commons/thumb/9/92/Neurons_uni_bi_multi_pseudouni.svg/330px-Neurons_uni_bi_multi_pseudouni.svg.png



1. WAS IST EIN NEURONALES NETZ?

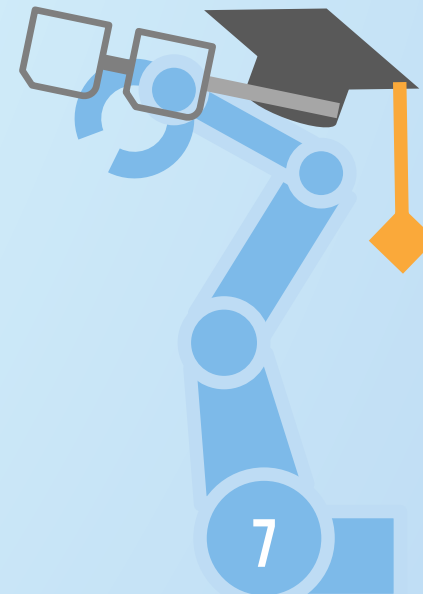
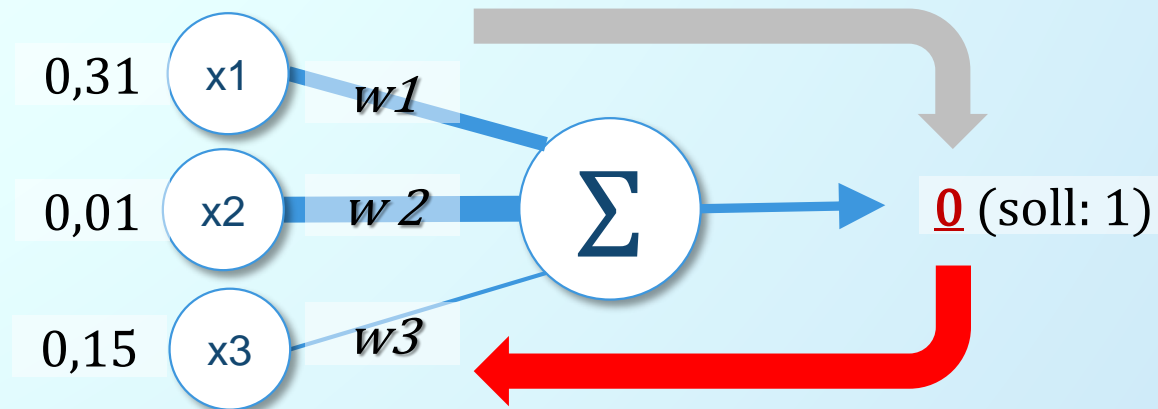
- Definierte Anzahl an Inputs
- Definierte Anzahl* an Outputs
- Beliebige viele Schichten (Layer)
- Ein Layer besitzt ein oder mehrere Neuron(en)
- Ausgabe: 0..1 (monoton steigend)



1. WAS IST EIN NEURONALES NETZ?

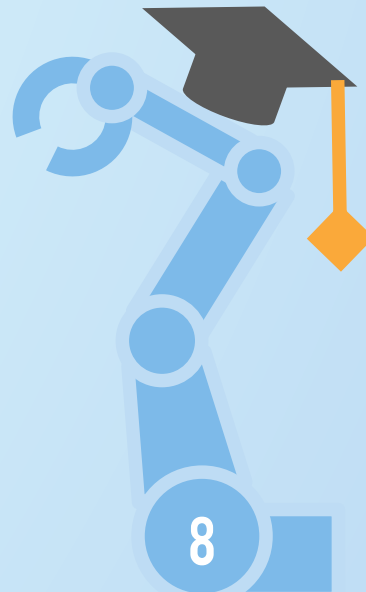
Trainingsphase – Testphase – Prediktion

- Unüberwachtes Lernen (unsupervised learning)
- Überwachtes Lernen (supervised learning)
 - Backpropagation



2. PROBLEMSTELLUNG

- Elektrisches Garagentor
- Selbsttätiges Öffnen bei Ankunft zuhause
- Funkstrecke vorhanden
- WLAN vorhanden
- Smart Home?



2. PROBLEMSTELLUNG

Frage nach der »einfachsten« Lösung

→ Lösung war nicht einfach

- Erkennung der **Handlung** des Nach-Hause-Kommens mit klassischen Algorithmen kaum möglich

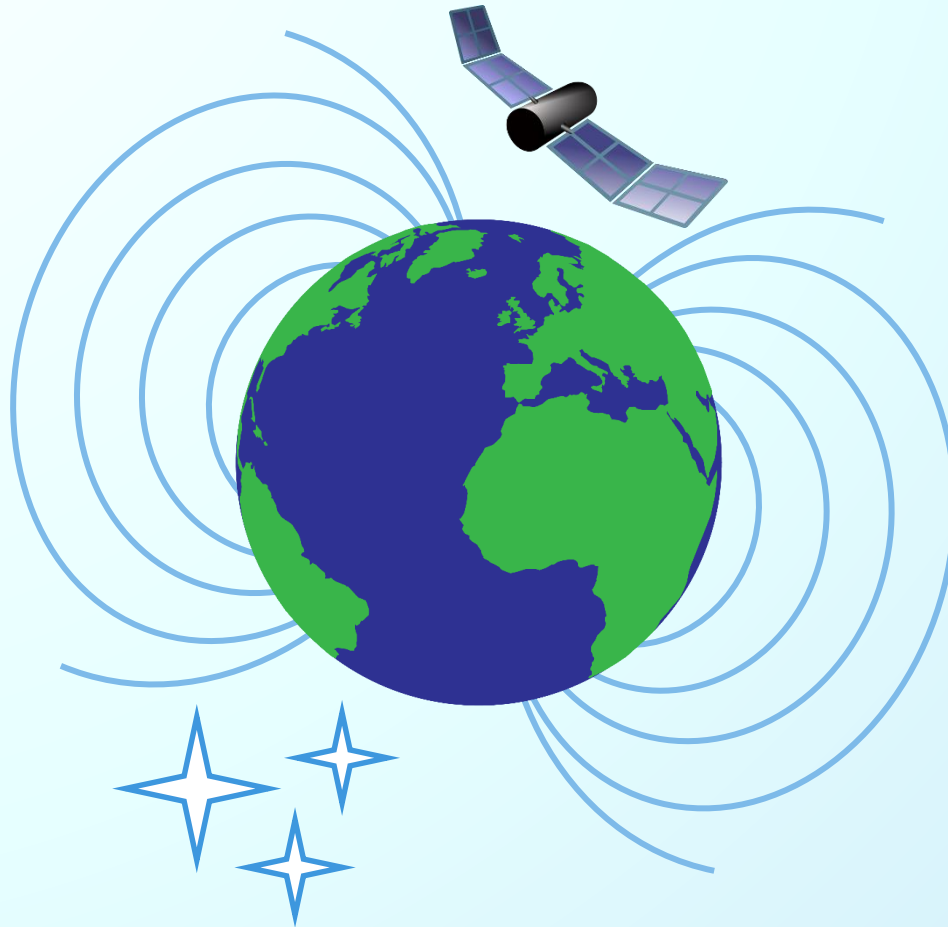
Problemzerlegung (Philosophie → Mathematik)

- Handeln \Leftarrow Zielgerichtet \Leftarrow Richtung \Leftarrow Vektor(en) \Leftarrow Zahl(en)
 - Es können also Vektor(en) (irgendwie) verarbeitet werden

Handeln = (...) von
Motiven geleitete **ziel-
gerichtete** Tätigkeit (...)
<https://de.wikipedia.org/wiki/Handeln>

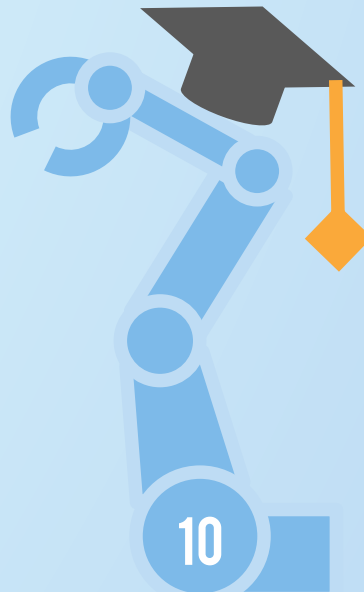


2. PROBLEMSTELLUNG – DATEN ERFASSEN



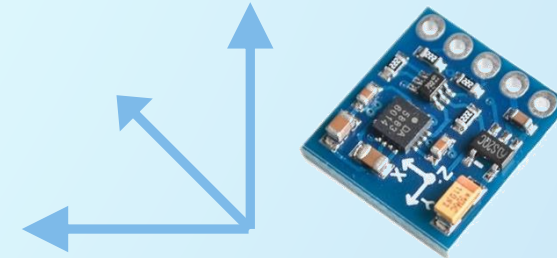
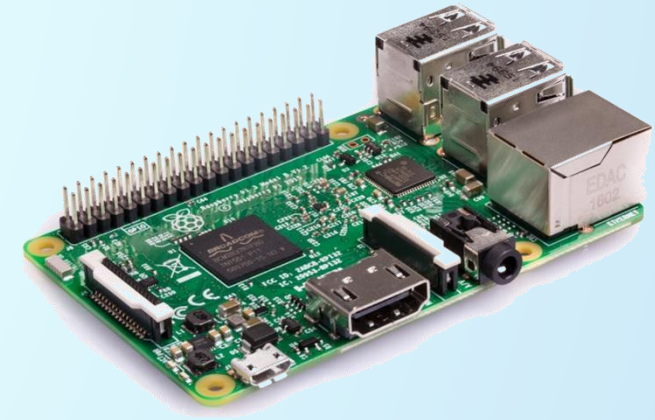
Die Bewegungsrichtung lässt sich messen:

- GPS
- Kompass
- Pulsare
(Röntgenstrahlung)
- Sternsensoren
- etliche mehr ...



3. HARDWARE / SOFTWARE (KOMPASS)

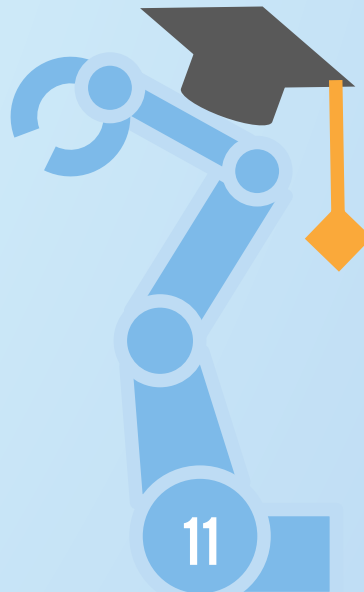
- Raspberry Pi 3
- Magnetfeldsensor QMC 5883 (3 Achsen)
- OS: Raspbian, Kernel 4.9.59-v7+ #1047
- Python 3 (3.5.3) Ausführung
- R 3.3.2 (x64)
- SQLite 3



Bildquellen:

<https://de.aliexpress.com/item/GY-271-HMC5883-HMC5883L-module-electronic-compass-compass-module-three-axis-magnetic-field-sensor/1556804905.html>

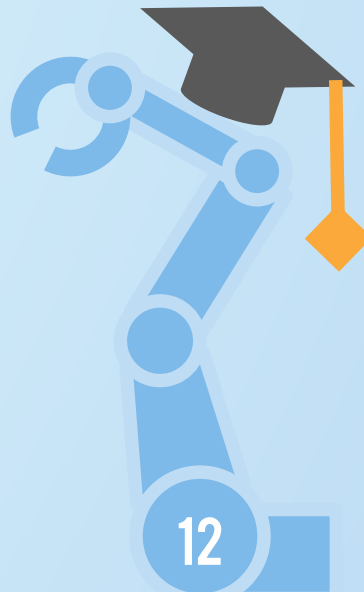
<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>



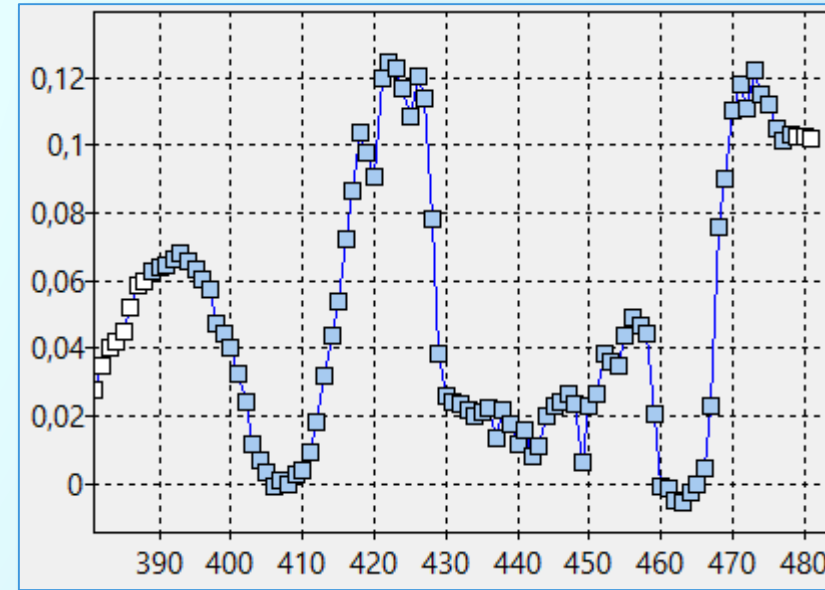
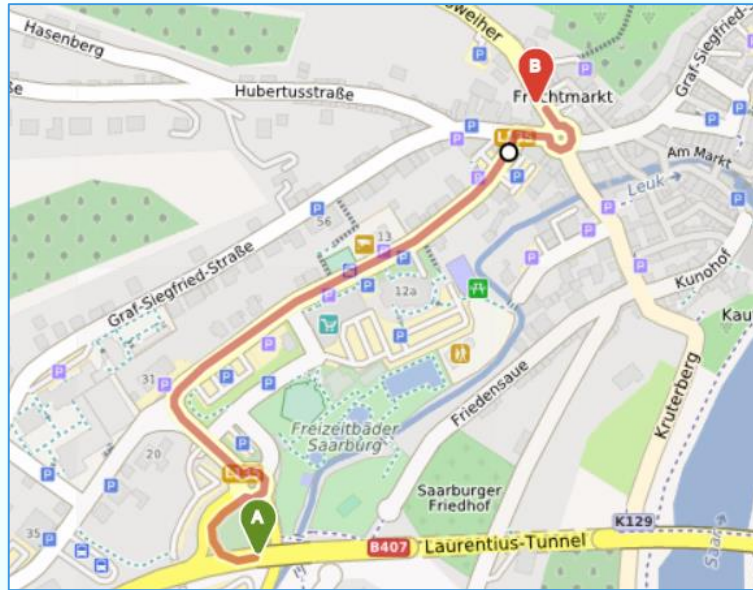
3. HARDWARE / SOFTWARE (KOMPASS)

Besonderheit des QMC 5883

- Ansteuerung in Python schwierig
- Verfügbarer Code im Netz war **nicht funktional**
- Python kennt zwei Datentypen: Zahl und Zeichenkette
- Rückgabewert Magnetfeldsensoren in zwei 8-Bit-Registern mit **negativem Vorzeichen ...**
 - Lösung musste erst programmiert werden und ist nun als Open Source im Netz verfügbar



4. DATEN SAMMELN, NORMALISIEREN, SPLITTEN



Bildquellen:
<http://maps.openrouteservice.org>
Eigene Software
»Ginger«

Wegstrecke in realer Welt

Messdaten, signifikantes Muster

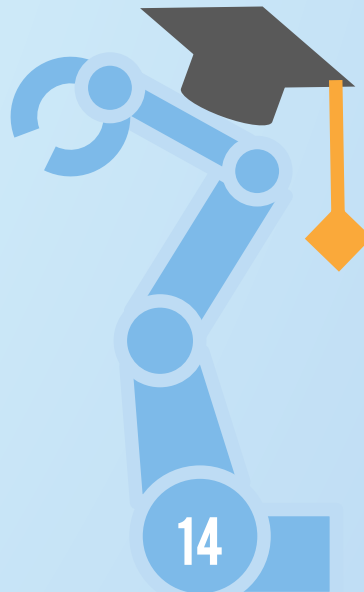
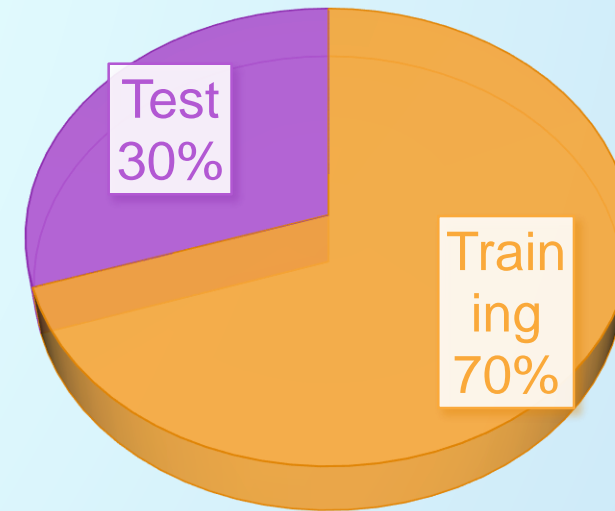
- Normalisierung bringt die Daten in Werte von 0 bis 1*
- Division aller Werte **einer Messreihe** durch das Maximum der Messreihe

4. DATEN SAMMELN, NORMALISIEREN, SPLITTEN

- Aufspalten der gesammelten Daten
- Trainingsdaten und Testdaten
- Trainingsdaten: ca. 70%
- Testdaten: ca. 30%

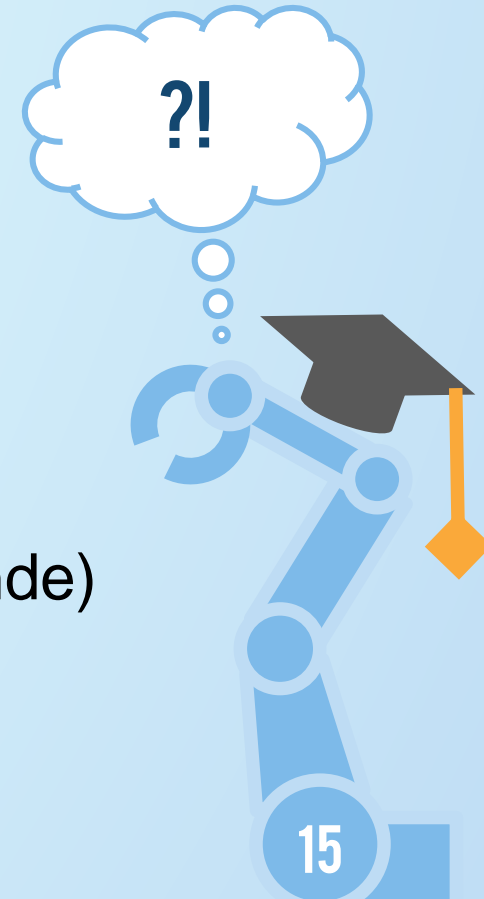
→ Ein Machine Learning-Modell ohne Evaluation sagt gar nichts aus!

ANTEIL



5. PRAXIS UND RESULTATE - ABLAUF

1. Daten sammeln auf dem Pi3 (Python, SQLite 3)
Export der Daten als `.db`-Datei
2. Training in R
Export der Gewichte des Netzes als `.csv`-Datei
3. Einsatz auf dem Pi 3 (Python)
Kein Training, sondern Laden der vortrainierten Gewichte
4. Permanente Magnetfeldmessung (1 Datensatz pro Sekunde)
Auswertung der letzten 80 Messwerte über das KNN
Signal bei Erkennung der *gerichteten Handlung*

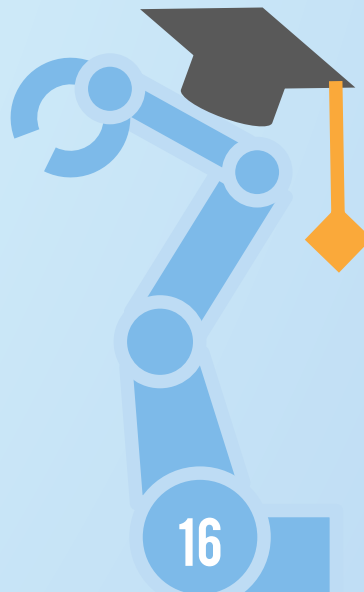


5. PRAXIS UND RESULTATE

Das resultierende künstliche neuronale Netz

- 80 Inputs \triangleq 80 Magnetfeld-Messpunkte (X-Achse, 1 pro Sekunde)
- 4 Layer mit {40, 20, 4, 1} Neuronen, ca. 4.084 Gewichte
- Übertragungsfunktion: Logistisch
- Daten: Ca. 992 Datensätze (63 positiv, 929 negativ)
- Trainingszeit in R: Ca. 5 min.

```
neuralnet(f, train, hidden = c(40, 20, 4), act.fct = 'logistic')
```



5. PRAXIS UND RESULTATE - EVALUATION

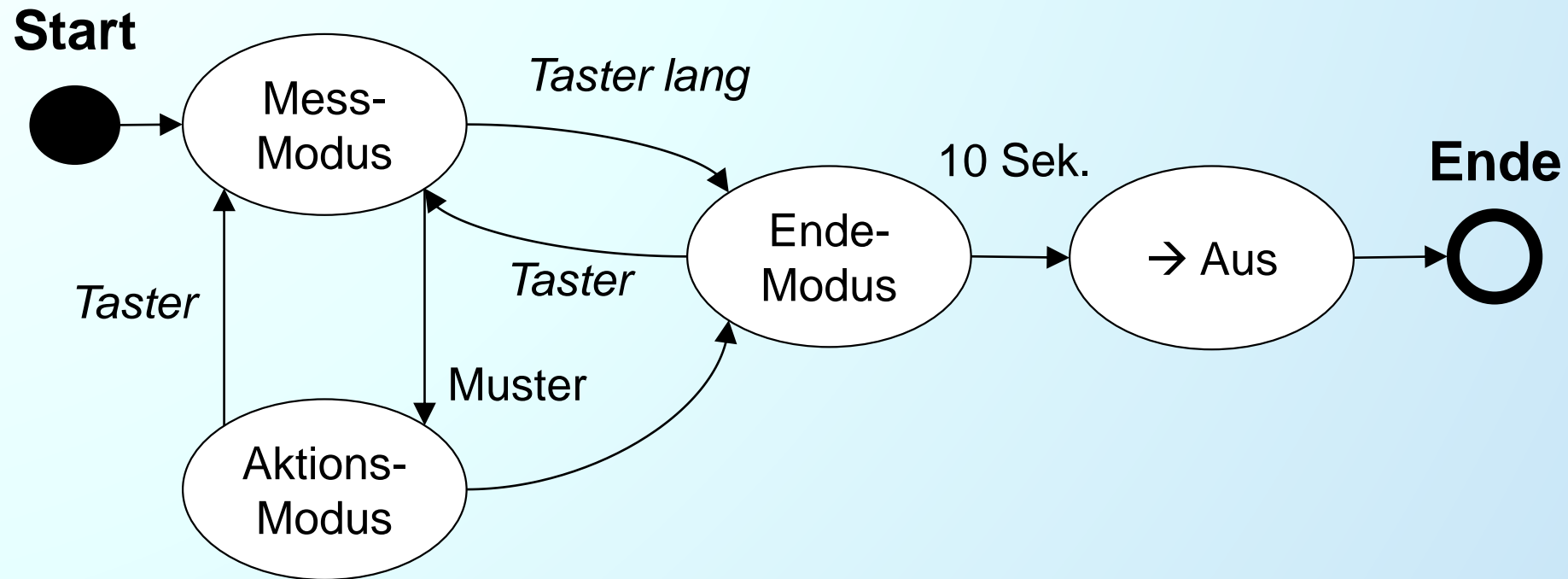
- Optimierung zugunsten des False Positive-Fehlers
- Treffergenauigkeit: 0,92 (92%)
- Falschklassifikationsrate: 0,08 (8%)

	Tatsächlich 1	Tatsächlich 0
Vorhersage 1	96	7
	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
Vorhersage 0	2	5
	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

$$\text{Treffergenauigkeit (Accuracy)} = \frac{\sum TP + \sum TN}{\sum TP + \sum FP + \sum FN + \sum TN}$$

5. PRAXIS UND RESULTATE - SOFTWARE

Software wurde als Zustandsautomat mit Python 3 implementiert



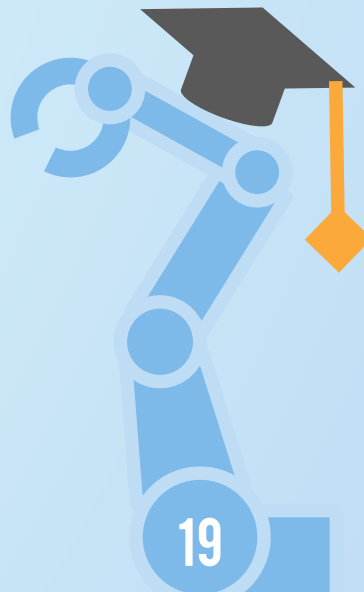
6. UNGELÖSTE PROBLEME

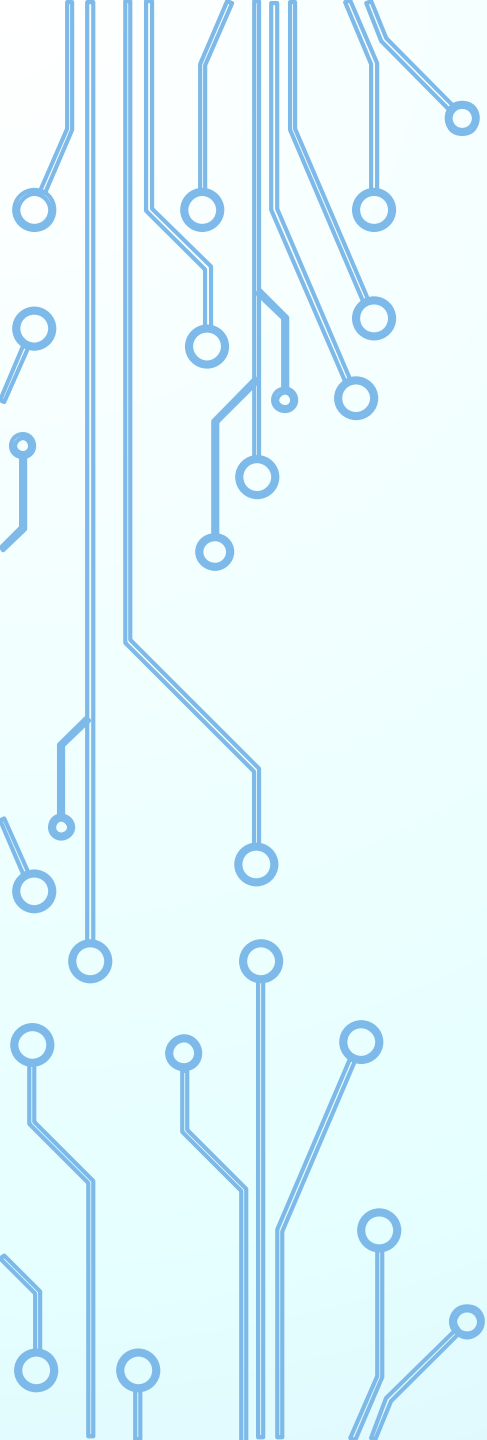
Hardware

- Messungenauigkeit (konnte durch Median-Bildung gemindert werden)

Eigenschaften des resultierenden Netzes

- Hohe Falsch-Positiv-Rate
- Datenmenge gering und teuer (1 Datensatz \triangleq 1 Autofahrt)
- Raspberry Pi in der Praxis zu langsam (echtzeitfähig?)





*Sänk You for
trävelling wis
Deutsche Bahn

VIELEN DANK*

Sebastian Folz, Neuronale Netze mit dem Raspberry Pi 3



20

03.11.2018