

PRAKTIKUM IS

Block Machine Learning

https://djcordhose.github.io/ai/2018_haw_ml_praktikum.html

Bewertung

- **Stufe 3:** 13 bis 15 Punkte
 - **Nicht-offensichtliche komplexe** Eigenschaften von Konzepten der künstlichen Intelligenz wurden erkannt
 - und zwar **ausreichend abstrakt und ausreichend verallgemeinert**.
 - Ein etwaiges Anwendungsumfeld wurde **korrekt, angemessen und effektiv** analysiert und modelliert.
- **Stufe 2:** 9 bis 12 Punkte
 - **Einzelne nicht-offensichtliche** Eigenschaften von Konzepten der künstlichen Intelligenz wurden erkannt.
 - Ein etwaiges Anwendungsumfeld wurde **korrekt** und **mit kleinen Einschränkungen angemessen** analysiert und modelliert.
- **Stufe 1:** 5 bis 8 Punkte
 - **Wenige recht offensichtliche** Eigenschaften von Konzepten der künstlichen Intelligenz wurden erkannt.
 - Ein etwaiges Anwendungsumfeld wurde **in einfacher Weise** so analysiert und modelliert, dass praktische Tests möglich sind.

Praktikum: Organisatorisches

- Ziel des Praktikums: etwas neues lernen
- Wir sind jeden Montag zu sprechen
 - von 10:00 - 12:00 (12.81) oder
 - 17:30 - 18:30 (Vorlesungsraum)
 - Fragen und Diskussion
 - Praktikumsabnahme

Letzter Abnahme-Termin für die Praktika

- ***Suche:*** KW21, 23.05.2018
- ***Lernen:*** KW24, 11.06.2018
- ***Natürliche Sprache,*** erste Prüfungen möglich: KW26, 25.06.2018

Prüfungstermine

- KW26,
25.06.2018
- KW27
- KW29

Update: Prüfungen

- Zwei Studierende haben ein gemeinsames Basisthema vorbereitet mit eigener, individueller Erweiterung.
- Die Basis wird gemeinsam vorgetragen, also jeder mit ca. 50% Redeanteil. (5 Minuten)
- Danach trägt jeder seine eigene Ergänzung vor. (2 x 5 Minuten).
- Gesamtzeit $4 \times 15 = 60$ Minuten.
- Die Ausarbeitungen sind individuell.
- Die Noten werden auch individuell vergeben.
- Einzelprüfung möglich, dann 7,5 bis 10 Minuten pro Thema.

Install ML Software

Local Installation:

- <https://www.anaconda.com/download/>
- **Important:** Install version 3.x
- git clone
git@github.com:DJCordhose/haw.git
- cd haw/notebooks
- jupyter notebook

Or clone on Azure Notebooks:

- <https://notebooks.azure.com/djcordhose/libraries/ml-haw>

Tolle Sammlungen von Notebooks

Als Inspiration oder Startpunkt für eigene Projekte

- <https://github.com/ageron/handson-ml>
- <https://github.com/fchollet/deep-learning-with-python-notebooks>



Mögliche Aufgaben für Praktikum

Beispiel Stufe 1: Wie funktionieren Waymos selbstfahrende Autos

Nachvollziehen und recherchieren

<small>

<https://www.theverge.com/2018/5/9/17307156/google-waymo-driverless-cars-deep-learning-neural-net-interview>

Classic ML

Beispiel Stufe 1: Random Forest und Adaboost anwenden

- Dasselbe wie bei Decision Trees für Random Forest und Ada Boost machen
- Parameter herausfinden
- Algorithmen verstehen

3-sklearn-decision-trees

Beispiel bis Stufe 3: Regularisierung auf Neuronalen Netzwerken

- Dropout: <https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/>
- BatchNormalizatiion: <https://towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c>
- L1 und L2: <https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a>
- Zum experimentieren mit L1 und L2: <http://playground.tensorflow.org>
- Allgemeine Hinweise: <https://machinelearningmastery.com/improve-deep-learning-performance/>

*Beispiel Stufe 2-3: Wende Machine
Learning auf eine eigene
Problemstellung an*

Beispiel bis Stufe 2: Erarbeite dir die Strategie Support Vector Machines

Nutze die Implementierung von sklearn und versuche einen besseren Wert zu erreichen als mit RandomForest

Deep Learning / CNNs

Beispiel bis Stufe 3: Ein Convolutional Neural Network mit eigenen Bilddaten trainieren

Aufwändig, da lokale installation oder tensorflow.js nötig und eigene Bilder besondere Herausforderung

- if you are looking for a handy python script for 'searching' and 'downloading' hundreds of Google images to the local hard disk for you next DL project, that's the one: <https://t.co/4nh3hnoZr1>
(<https://twitter.com/rasbt/status/982822969947848704?s=03>)
- Pokemon Images:
<https://www.pyimagesearch.com/2018/04/16/keras-and-convolutional-neural-networks-cnns/>

Hinweise: Ein Convolutional Neural Network mit eigenen Bilddaten trainieren

- Man braucht mind. 5000 Bilder pro Kategorie
- Das hat man oft gar nicht. Dann kann man künstlich Variationen erzeugen
- passende Architektur wählen (eigene, z.B. ResNet, ImageNet)
- Evtl. auf ImageNet vortrainierte Modelle nehmen und nur die letzten Layers nachtrainieren
- Auch mit sehr guter Hardware dauern Experimente Stunden
- Mit Teildaten Overfitten, um zu sehen, ob das Modell überhaupt genug Kapazität hat

Beispiel bis Stufe 3: Object Detection

<https://www.datacamp.com/community/tutorials/object-detection-guide>
<https://towardsdatascience.com/deep-learning-in-your-browser-a-brisk-guide-ca06c2198846>
<https://github.com/ModelDepot/tfjs-yolo-tiny>

Beispiel bis Stufe 2: Experimente, wie funktionieren CNNs?

For instance, by combining feature visualization (*what is a neuron looking for?*) with attribution (*how does it affect the output?*), we can explore how the network decides between labels like **Labrador retriever** and **tiger cat**.



Several floppy ear detectors seem to be important when distinguishing dogs, whereas pointy ears are used to classify "tiger cat".

CHANNELS THAT MOST SUPPORT ...

LABRADOR RETRIEVER

TIGER CAT

[feature visualization](#) of channel

hover for attribution maps →

net evidence

1.63

1.51

1.19

1.32

1.54

1.72

for "Labrador retriever"

1.22

1.24

1.32

-0.70

-1.24

-0.43

for "tiger cat"

-0.40

-0.27

0.13

0.62

0.30

1.29

<https://distill.pub/2018/building-blocks/>

Beispiel Stufe 1: Neural Network tunen

- Netzwerk für Problemstellung optimieren.
- Kannst du es verbessern?
- Woran erkennst du eine Verbesserung?
- Wie klein kann man das Netzwerk machen?

4-tf-keras-nn / 4-keras-tensorflow-nn

Reinforcement Learning

Beispiel bis Stufe 3: Experimente aus dem Open AI gym

Python Kenntnisse notwendig, alle Schwierigkeitsstufen denkbar

https://gym.openai.com/envs/#classic_control

Beispiel bis Stufe 3: Ein eigenes Browserspiel mit DQNs (Reinforcement Learning bauen)

Projekt braucht kein Setup, man sollte JavaScript-
Kenntnisse haben

<http://web.sfc.keio.ac.jp/~t15704yn/falling/index.html>

https://github.com/seann999/dodge_tfjs

Tutorial: Write an AI to win at Pong from scratch with Reinforcement Learning.

- Unklar welche Stufe, für die Prüfung müsstet ihr euch da noch etwas ausdenken

<https://medium.com/@dhruvp/how-to-write-a-neural-network-to-play-pong-from-scratch-956b57d4f6e0>

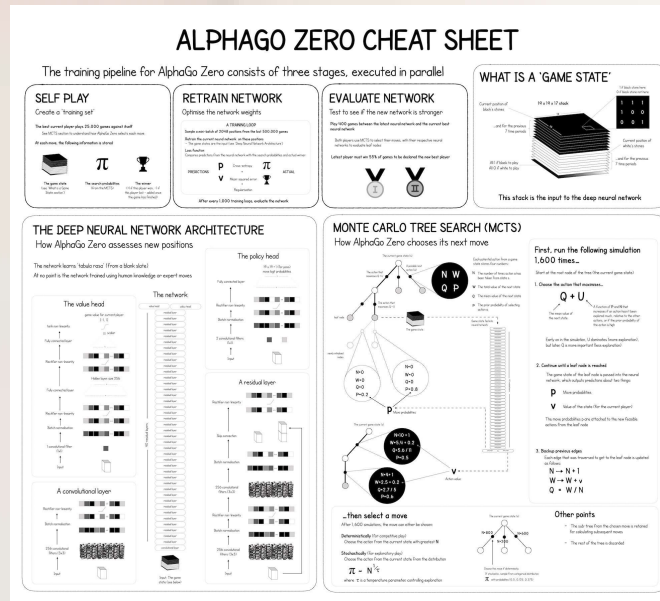
Beispiel bis Stufe 3: Atari spielen lernen auf dem eigenen PC

aufwändig, lokale Installation und GPU notwendig, aber Code und Blogpost sind vorhanden

Post: <https://eng.uber.com/accelerated-neuroevolution/>

Code: https://github.com/uber-common/deep-neuroevolution/tree/master/gpu_implementation

Beispiel bis Stufe 3: AlphaGo Zero nachbauen



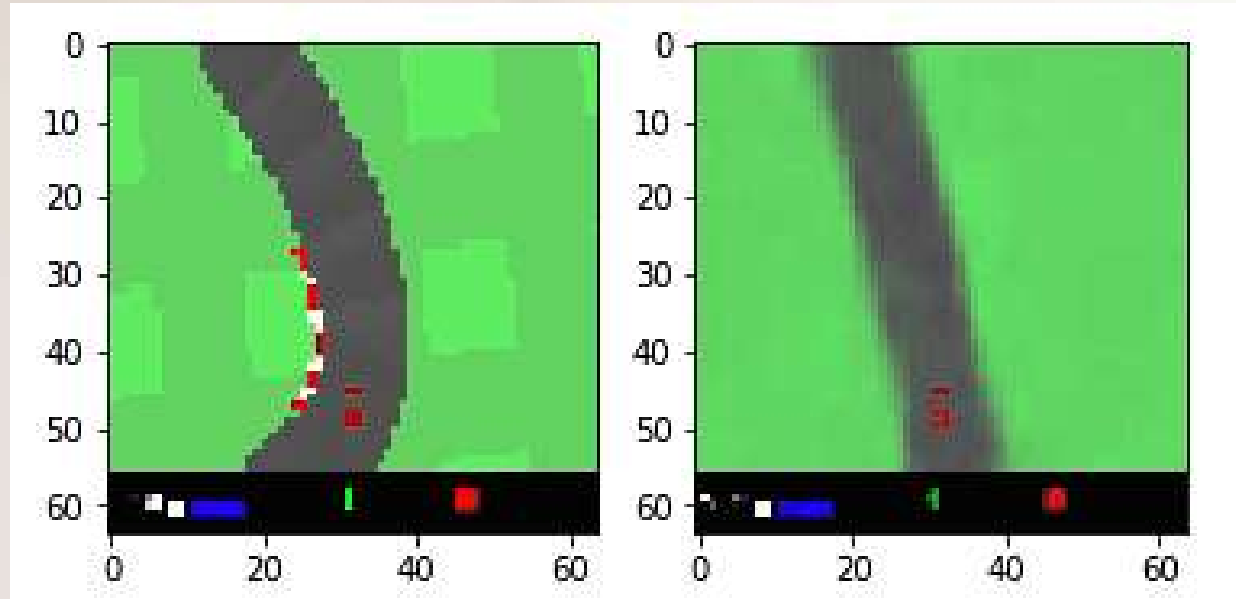
aufwändig, lokale Installation notwendig, aber Code und Tutorial ist vorhanden

Tutorial: <https://medium.com/applied-data-science/how-to-build-your-own-alphazero-ai-using-python-and-keras-7f664945c188>

Code: <https://github.com/AppliedDataSciencePartners/DeepReinforcementLearning>

Technische Erklärung: <http://tim.hibal.org/blog/alpha-zero-how-and-why-it-works/>

Beispiel bis Stufe 3: World Model nachbauen



aufwändig, lokale Installation notwendig, aber Code und Tutorial ist vorhanden

Tutorial: <https://medium.com/applied-data-science/how-to-build-your-own-world-model-using-python-and-keras-64fb388ba459>

Code: <https://github.com/AppliedDataSciencePartners/WorldModels>

Unsupervised Learning

*Beispiel bis Stufe 3: UMAP selbst
erarbeiten und erklären*

Beispiel bis Stufe 2: Finde ein eigenes Beispiel für Clustering und probiere unterschiedliche Strategien aus

- mit den Parametern der bekannten Algos experimentieren
- mit welchen Parametern
- Clustering. Anderer Algo, Birch.

Notebook: k-means_vs_dbscan, plot_cluster_comparison

Beispiel bis Stufe 3: Implementiere einen eigenen Clustering Algorithmus

- den Algorithmus sollte es so noch nicht geben
- wie unterscheidet er sich von DBSCAN und k-means?