

Themenideen für Abschlussarbeiten (Bachelor- und Masterarbeiten)

Brainstorming

Prof. Dr.-Ing. Jürgen Brauer

Zuletzt aktualisiert: 07.06.19

Meine Interessensgebiete liegen in den Bereichen:

- Übertragung von biologisch motivierten Lernansätzen auf maschinelles Lernen (ML)
- Maschinelles Lernen für Fahrerassistenzsysteme
- Bildverarbeitung
- Robotik

1. Masterarbeit ML: „Lokale Lernregeln für CNNs“	1
2. Bachelor- oder Masterarbeit ML: „Optimierung des Lernvorgangs von Neuronalen Netzen“	2
3. Bachelor- oder Masterarbeit ML: „Neuronale Netze mal anders!“	2
4. Bachelor- oder Masterarbeit ML: „Neural Architecture Search (NAS)“	3
5. Bachelor- oder Masterarbeit ML: „Zufällige Neuronale Netze“	3
6. Masterarbeit ML: „Deep Learning für Fahrerassistenzsysteme“	3
7. Bachelor- oder Masterarbeit Robotik: „Ein Roboter lernt das Greifen“	3
8. Bachelor- oder Masterarbeit Robotik: „Ein Roboter lernt das Laufen“	4

1. Masterarbeit ML: „Lokale Lernregeln für CNNs“

Das CNN ist heute sicherlich einer der wichtigsten Deep-Learning Modelle. Das CNN besteht aus einer Merkmalshierarchie und einem anschließenden MLP als Klassifikator. Das Model wird heute „top-down“ mittels des „Backpropagation“-Algorithmus „überwacht“ trainiert, d.h., es müssen Beispiele (x,y) für die Trainingsphase erstellt werden, die dem Modell zeigen, wie Inputs x auf Outputs y abzubilden sind. Mit Hilfe der „Targets“ y werden dann die Gewichte so eingestellt, dass die Inputs x möglichst gut auf die Outputs y abgebildet werden. Beispiel aus dem Bereich der Bildklassifikation: x =Eingabebilder von Katzen und Hunden, $y=(1,0)$ (bedeutet „Katze“) oder $y=(0,1)$ (bedeutet „Hund“).

Da das Erstellen der Trainingsbilder sehr aufwändig ist, würde man gerne die Merkmalshierarchie des CNNs „überwacht“, d.h. nur durch Trainingsinputs x trainieren.

In der Arbeit soll daher untersucht werden, welche heutigen Ansätze es gibt, um CNNs bereits unüberwacht nur durch Hilfe sog. „lokaler Lernregeln“ zu trainieren (Literaturarbeit! Ca. 1/3 der Arbeit). Lokale Lernregeln betrachten nur die Aktivitäten benachbarten Neurone und lernen die Gewichte ohne die Verwendung von „Targets“. Deep Learning Experten wie Geoffrey Hinton sehen in diesem „unüberwachten“ Lernansatz die Zukunft des Gebiets „Deep Learning“. Die Merkmalshierarchie kann dadurch „bottom-up“ trainiert werden.

Dann soll einer der Ansätze ausgewählt, implementiert und experimentell untersucht werden (praktischer Anteil, ca. 2/3 der Arbeit). Idealerweise werden hier auch eigene Ideen (z.B. kleinere Modifikationen des gewählten Ansatzes) ausprobiert.

2. Bachelor- oder Masterarbeit ML: „Optimierung des Lernvorgangs von Neuronalen Netzen“

Heute werden in jeder „Epoche“ einem Neuronalen Netz beim Training jedes der Trainingsbeispiele einmal „gezeigt“. Dabei wird die Reihenfolge der Präsentation der Trainingsbeispiele typischerweise zufällig erstellt („shuffling of the training data“).

Was gibt es bereits für Ansätze um hier den Lernvorgang zu optimieren (Optimierungsziel: kürzere Trainingszeit bei gleicher Performanz des NN oder evtl. sogar bessere Performanz)? Dazu soll eine Literatursuche durchgeführt werden (1/3 der Arbeit).

Dann soll ein Ansatz ausgewählt und experimentell untersucht werden (2/3 der Arbeit).

Beispiel für einen Ansatz: *Aufmerksamkeitssteuerung bei der Zusammensetzung der Trainingsdaten für die nächste Epoche: nehme nur solche Trainingsdaten in die nächste Epoche, bei denen der aktuelle Fehlers des NN noch groß ist* (Intuition: ein Schüler / Student konzentriert sich bei dem Lernen auf diejenigen Teile des Stoffes, die er noch nicht verstanden / „gut kann“ und lässt die bereits erlernten Teile des Stoffes weg).

3. Bachelor- oder Masterarbeit ML: „Neuronale Netze mal anders!“

Die heutige Forschung und Praxis der Neuronalen Netze nutzt weitestgehend immer das gleiche technische Neuronenmodell, die sog. „Perzeptron Neurone“ (Ausnahme sind die „LSTM Cells“). Diese Perzeptron-Neurone berechnen die gewichtete Summe der Eingangssignale (=Aktivität des Neurons) und bilden diese „Aktivität“ auf einen Output mit Hilfe einer „Aktivierungsfunktion“ ab.

Dieser Trend der Verwendung des gleichen Neuronenmodells hat sich durch die breite Verwendung von Deep Learning Bibliotheken wie „TensorFlow“, „Keras“, „Caffe2“, etc. auch noch verstärkt, da man hier meist nur Standardschichten mit diesen Standardneuronen zu konkreten Modellen „zusammensteckt“. Die Bibliotheken bieten schlichtweg nur dieses einzige (allerdings auch sehr erfolgreiche) Neuronenmodell an und meist wird dies auch nicht hinterfragt oder als „Hyper-Parameter“ angesehen.

In der Bachelor- oder Masterarbeit sollen daher mal Netze bestehend aus neuartigen Neuronen (Basisrecheneinheiten) ausprobiert werden. Hierbei sollen zuerst Ideen gesammelt werden, wie die Neuronen anders im Rechner modelliert werden könnten als Standard-Perzeptron-Neurone. Dazu kann auch ein Blick auf das biologische Vorbild, die biologischen Neurone, gewagt werden, um sich Inspirationen abzuholen.

Beispielansatz: Aktuelle technische Neurone arbeiten deterministisch. Wie verhält es sich mit probabilistisch feuernenden Neuronen? Hat vielleicht die Verwendung von sog. „Monte-Carlo Neuronen“ Vorteile beim Kampf gegen das Overfitting (=Überanpassung des Modells auf die Trainingsdaten)?

Der nächste Schritt ist die Entwicklung von Lernregeln für diese Neurone. Auf einem einfachen Benchmark wie z.B. der Klassifikation sollen dann die Neuronenvarianten ausprobiert und evaluiert werden.

4. Bachelor- oder Masterarbeit ML: „Neural Architecture Search (NAS)“

In der Praxis stehen „Data Scientists“ bei der Verwendung von Neuronalen Netzen bei konkreten Anwendungsproblemen immer wieder vor dem gleichen Problem: Wie sollen die Hyper-Parameter des Modells denn jetzt eingestellt werden? Hyper-Parameter sind alle Modellparameter, die direkt vorgegeben werden und nicht durch die Trainingsdaten mitgelernt werden, z.B. bei einem MLP: Anzahl der Schichten und Anzahl der Neurone pro Schicht und bei einem CNN: Anzahl der Filterschichten, Anzahl der Filter pro Schicht, Filtergrößen, Filterstride, etc.

Unter „Neural Architecture Search“ (oder „AutoML“) versteht man den neueren Ansatz diese Suche nach guten Hyperparametern zu automatisieren.

In der Arbeit soll zuerst eine Literatursuche zum Thema NAS durchgeführt werden (1/3 der Arbeit), dann ein Standardverfahren implementiert und mit einer eigenen Idee verglichen werden (2/3 der Arbeit).

5. Bachelor- oder Masterarbeit ML: „Zufällige Neuronale Netze“

Es gibt einige wenige Veröffentlichungen, die zeigen, dass auch ein CNN mit zufällig initialisierten Filtergewichten, bei dem nur das MLP (der Klassifikator) trainiert wird, gute Klassifikationsleistungen zeigen kann. Ein solches Netz, bei dem die Merkmals-hierarchie nicht gelernt sondern nur zufällig initialisiert wurde, kann damit als „Baseline“ (Vergleich) für Lernverfahren dienen, bei denen die Merkmals-hierarchie gelernt wird.

In der Arbeit soll das Thema der zufälligen Initialisierung bzw. allgemein des Zufalls für NN untersucht werden. Beginnend mit einer Literaturarbeit (1/3 der Arbeit) sollen Experimente mit zufälligen NN durchgeführt werden (2/3 der Arbeit).

6. Masterarbeit ML: „Deep Learning für Fahrerassistenzsysteme“

Es soll ein System entwickelt werden, das die Fahrspur- als auch andere Verkehrsteilnehmer in Kamerabildern erkennt. Hierzu bieten sich aktuelle Objektdetektoren wie Mask-RCNN und Faster R-CNN an. Das entwickelte System soll so vorbereitet werden, dass es in Zukunft für die Lehre im Studiengang Fahrerassistenzsysteme eingesetzt werden kann (als Python-Code).

7. Bachelor- oder Masterarbeit Robotik: „Ein Roboter lernt das Greifen“

Am Anfang soll eine Hardware für einen (sehr) einfachen Greifroboter selber gebaut oder aber eingekauft werden. Für diesen Greifer soll dann ein biologisch motivierter Lernprozess des Greifens implementiert werden.

Das Standard Reinforcement-Learning-Framework (Lernen durch Versuch- und Irrtum) kann hier als möglicher Ausgangspunkt genutzt werden. Wie aber schafft man es, dass der Roboter in akzeptabler Zeit das Greifen eines beliebigen Objektes (Greifer und Objekte auf einem Tisch werden durch eine Kamera beobachtet) erlernen kann?

8. Bachelor- oder Masterarbeit Robotik: „Ein Roboter lernt das Laufen“

Ein humanoider Roboter (zweibeiniger Roboter mit zwei Armen, Kopf, Rumpf) in einer Simulationsumgebung wie dem „Gym“ von OpenAI [1] soll selbständig die Fortbewegung mittels biologisch motivierten Lernen erlernen.

Hierbei sollen zwei Teilziele untersucht werden:

1. Der Roboter darf sich irgendwie fortbewegen (auch unter Verwendung der Arme, etc.) und
2. Der Roboter soll sich auf zwei Beinen laufend fortbewegen

Man kann sich hier an aktuellen Arbeiten hierzu von OpenAI orientieren oder komplett neue Ansätze im Rahmen der Arbeit ausprobieren.

Eine solide Literatarbeit (1/3 der Arbeit) soll der experimentellen Arbeit (2/3 der Arbeit) auch hier wieder vorangehen.

[1] <https://gym.openai.com/envs/Humanoid-v2/>