BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Raum-zeitliche Mustererkennung in Sensordaten mit Deep Learning

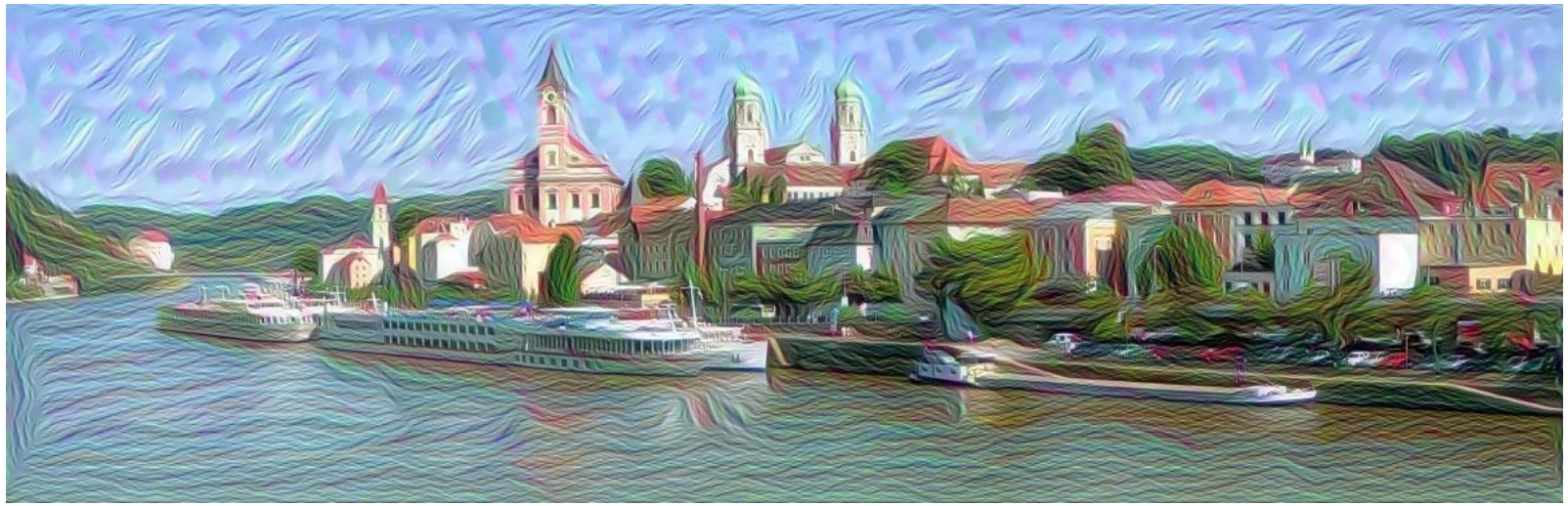
Felix Kunde, Petra Sauer

About me

- Wissenschaftlicher Mitarbeiter @BeuthHS
- Geographie / GIS Background
- Fokus der letzten Jahre: Geodatenbanken
- Core Contributor für 3D City Database (CityGML) und pgMemento (PostgreSQL Versioning)
- Twitter: @FIxKu , GitHub: FxKu

Gliederung

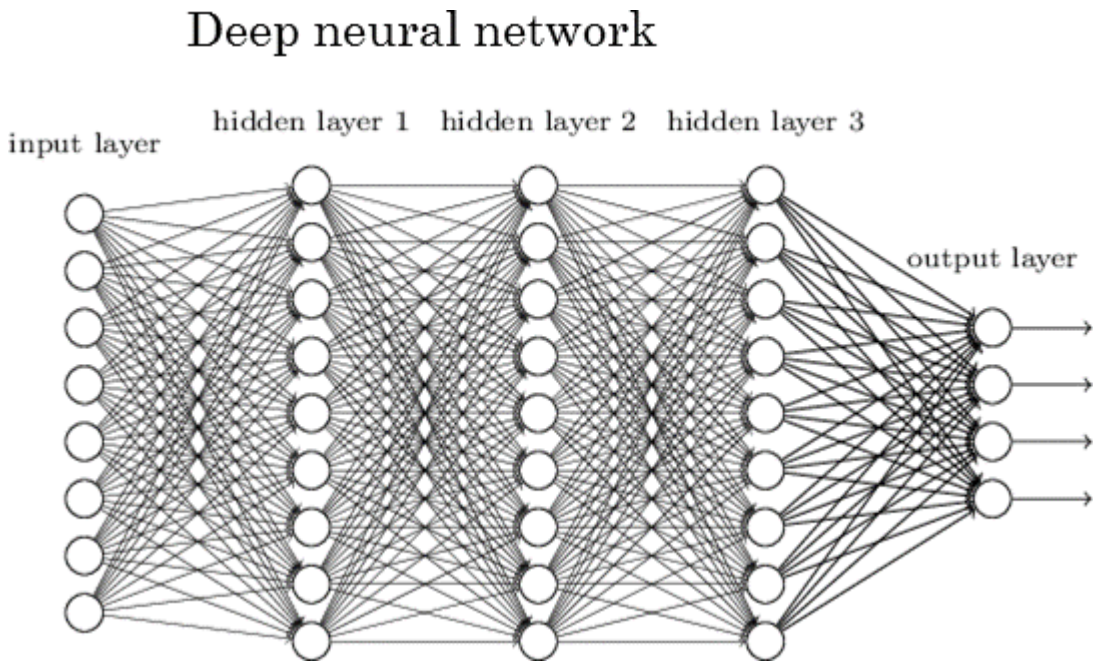
- **Was ist Deep Learning?**
  - Was steckt hinter dem Hype?
- **Geht das auch mit Geodaten?**
  - Ein paar Beispiele
- **Worauf muss man achten?**
  - Wie viel Theorie braucht man?
- **Wie fange ich an?**
  - Welche Software gibt es?

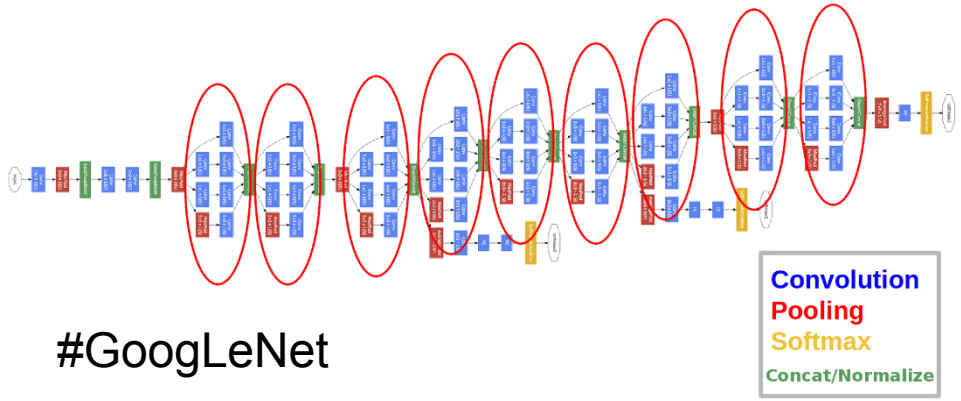


#DeepDream

Deep Learning Demystified

- Deep Learning steht für maschinelles Lernen mit künstlichen neuronalen Netzen – engl. artificial neural network (**ANN**)
- Ein ANN beschreibt ein **Netz aus mathematischen Funktionen**, über dass sich **Signale** propagieren lassen
- Ein ANN besteht in der Regel aus verschiedenen **Schichten** (Layern) von "**Neuronen**", die untereinander mit gerichteten **Kanten** verbunden sind
- Das Wort "**Deep**" soll zum Einen die Tiefe der Netze hervorheben. Zum Anderen ist es einfach nur ein Buzzword. Ähnlich wie A.I.





#GoogLeNet

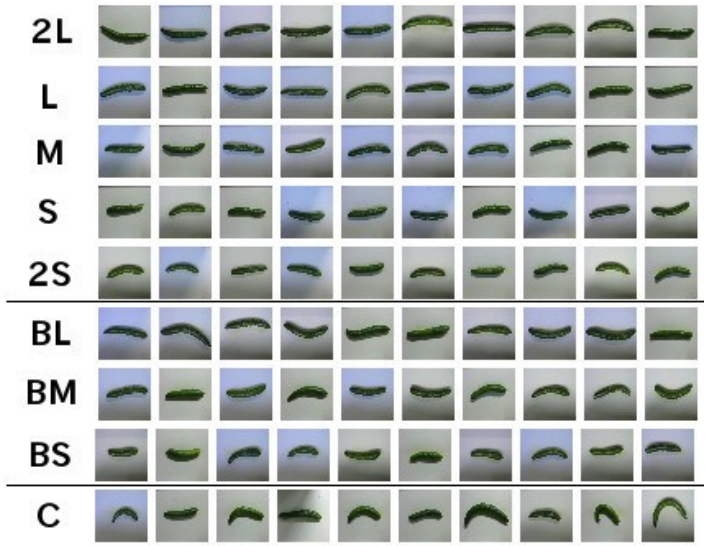
Deep Learning: Was ist dran am Hype?

#DeepMind #AlphaGo

Deep Learning: Was ist dran am Hype?

- Google Translate
- Bilderkennung
- Krebserkennung
- Text Mining, Suchmaschinenoptimierung, Chatbots
- Spracherkennung (z.B. Siri, Watson)
- Selbstfahrende Autos

https://cloud.google.com/blog/big-data/2016/08/how-a-japanese-cucumber-farmer-is-using-deep-learning-and-tensorflow



#Nvidia #Tesla

#Sunspring

Deep Learning: Funktionsweise

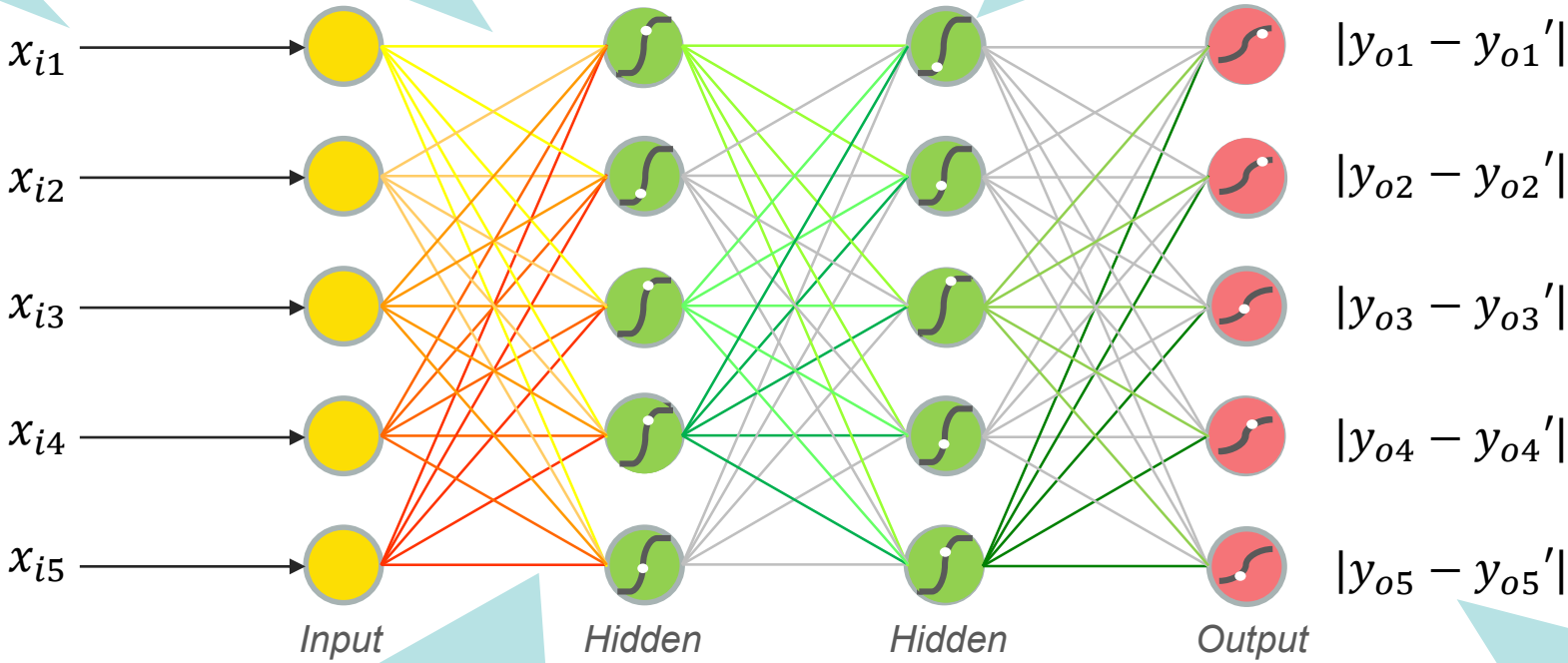
Eingangsdaten werden **normalisiert** (0 - 1) und gehen als **Vektoren** ein

Kanten besitzen **Gewichte**, die mit den Werten aus den Neuronen **multipliziert** werden

Eine **Schwellwertfunktion** bestimmt, ob die eingehenden Werte das Neuron **aktivieren**

Errechnung des Abweichung (**Fehler**) zu korrekten Wert

INPUT
Ein Pixel
Ein Frame
Eine Frequenz
Ein Wort
Eine Silbe
Ein Messwert
Eine Aktion
…

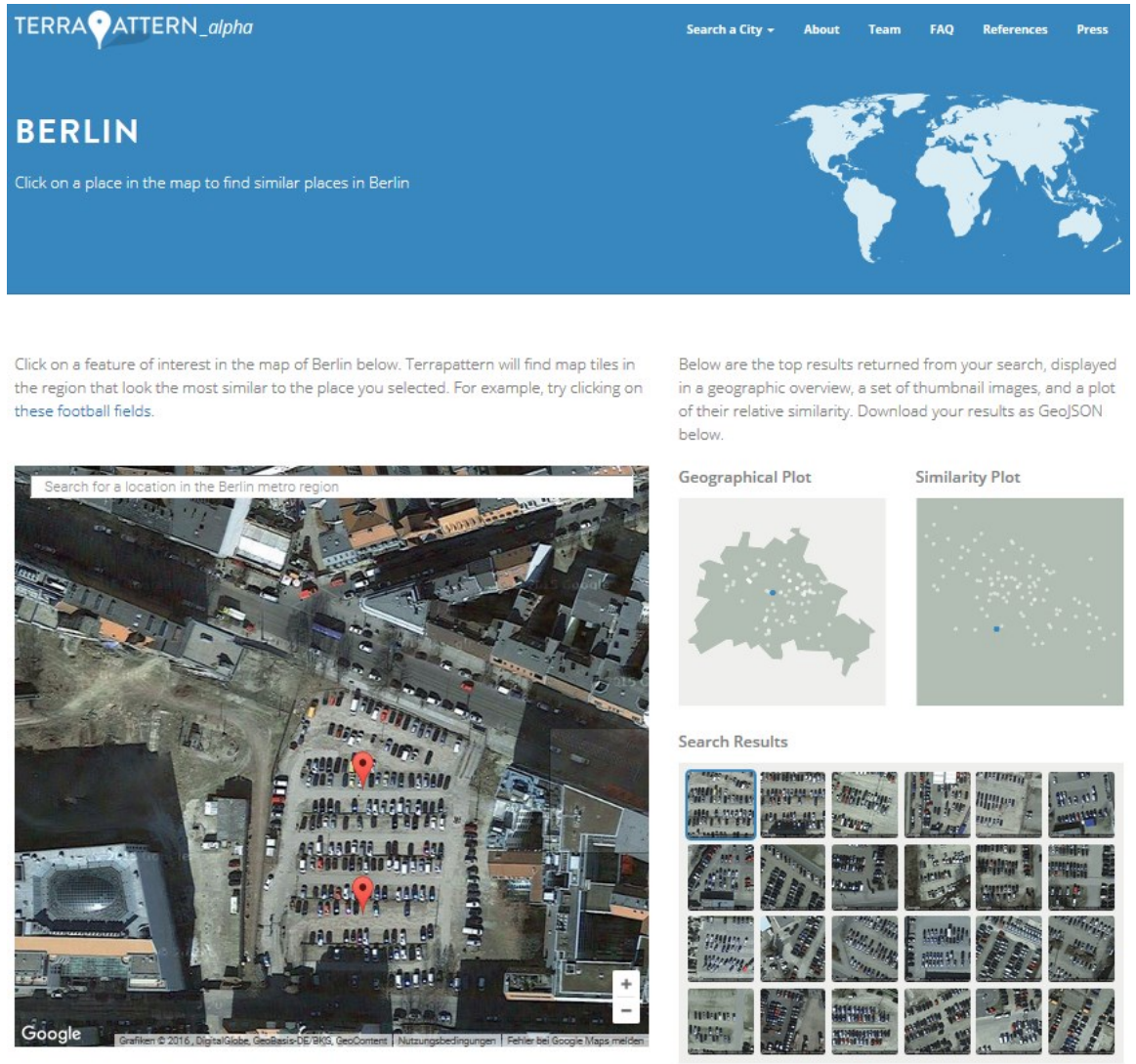


Merkmalsvektor vom zu bestimmenden Ziel (***TARGET***)

Der „Fehler“ wird **zurück** durch das Netz **propagiert**. Die Gewichte der Kanten werden entsprechend ihre Stärke angepasst

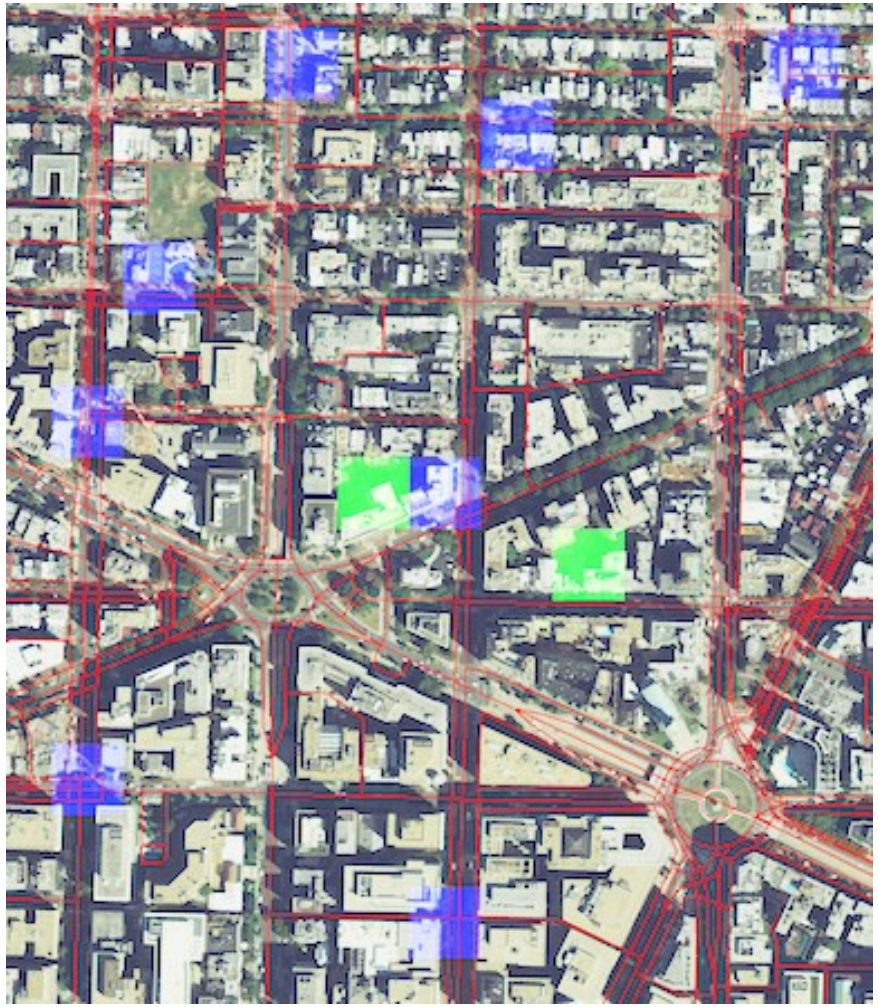
Suche nach optimaler Kombination von Gewichten für den geringsten Fehler ist **NP-Hard**. Stattdessen wird der Fehler Stück für Stück **minimiert** werden, bis ein **Minimum** erreicht wird.

Geht das auch mit Geodaten?

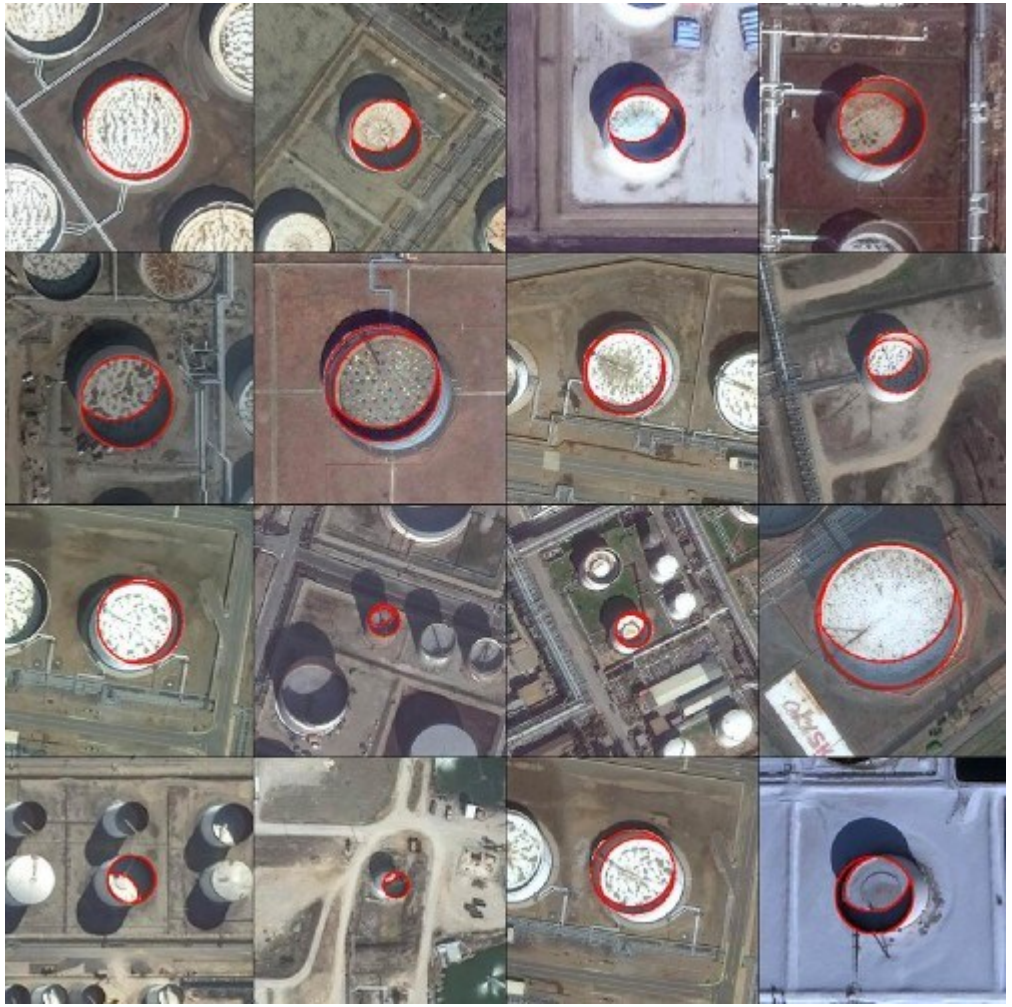


#Terrapattern

Geht das auch mit Geodaten?



#DeepOSM



#Orbitalinsight

Deep Learning auf Zeitreihen

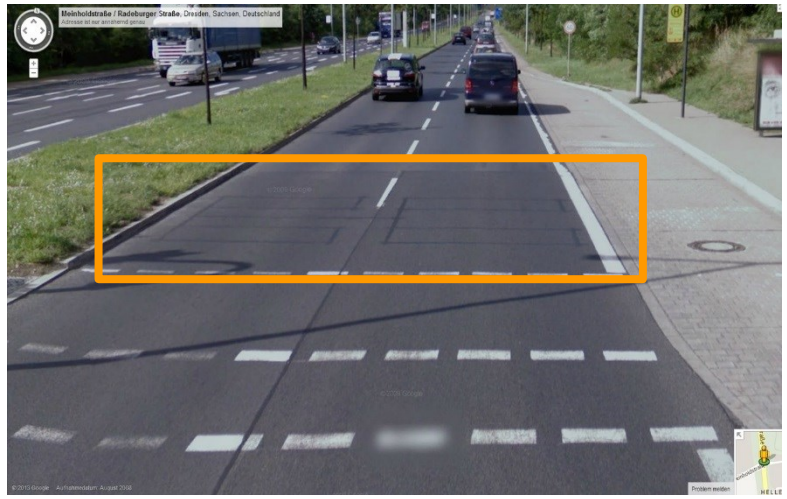
- Es gibt viele verschiedene Algorithmen für ANNs. Beliebt sind u.a. **Convolutional** NNs, die spezielle **Extraktionsschichten** besitzen, oder **rekurrente** NNs, die ein „**Gedächtnis**“ besitzen und **Sequenzen** lernen können

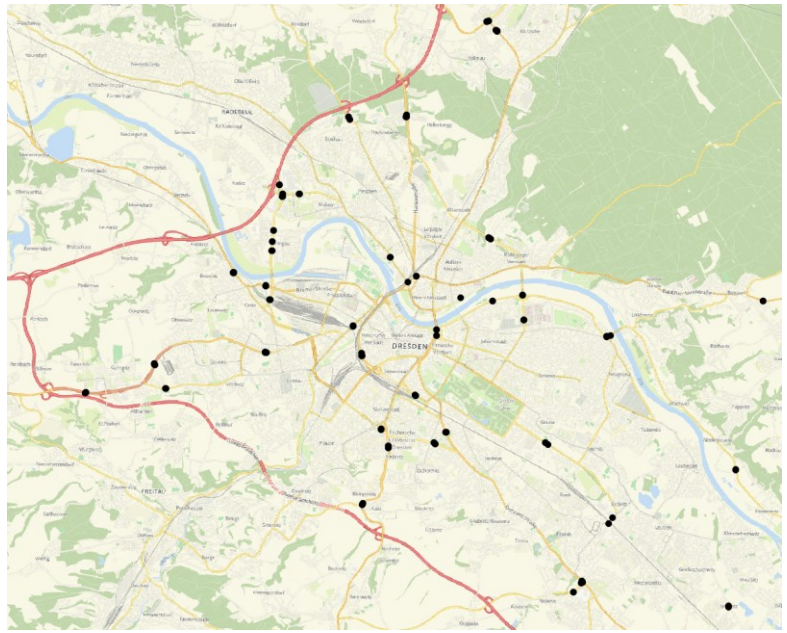
- **Idee**: RNNs müssten sie ja eigentlich zeitliche Muster in Messreihen erkennen können

- Anwendungsfall in Forschungsprojekt:
  - 8 Jahre Messreihen von stationären Verkehrssensoren aus Dresden (VAMOS)
  - Kurzfristprognosen werden in der Praxis nicht eingesetzt bzw. ihnen wird misstraut
  - Implementierung einer Vorhersage für 5 Minuten bis 1 Stunde

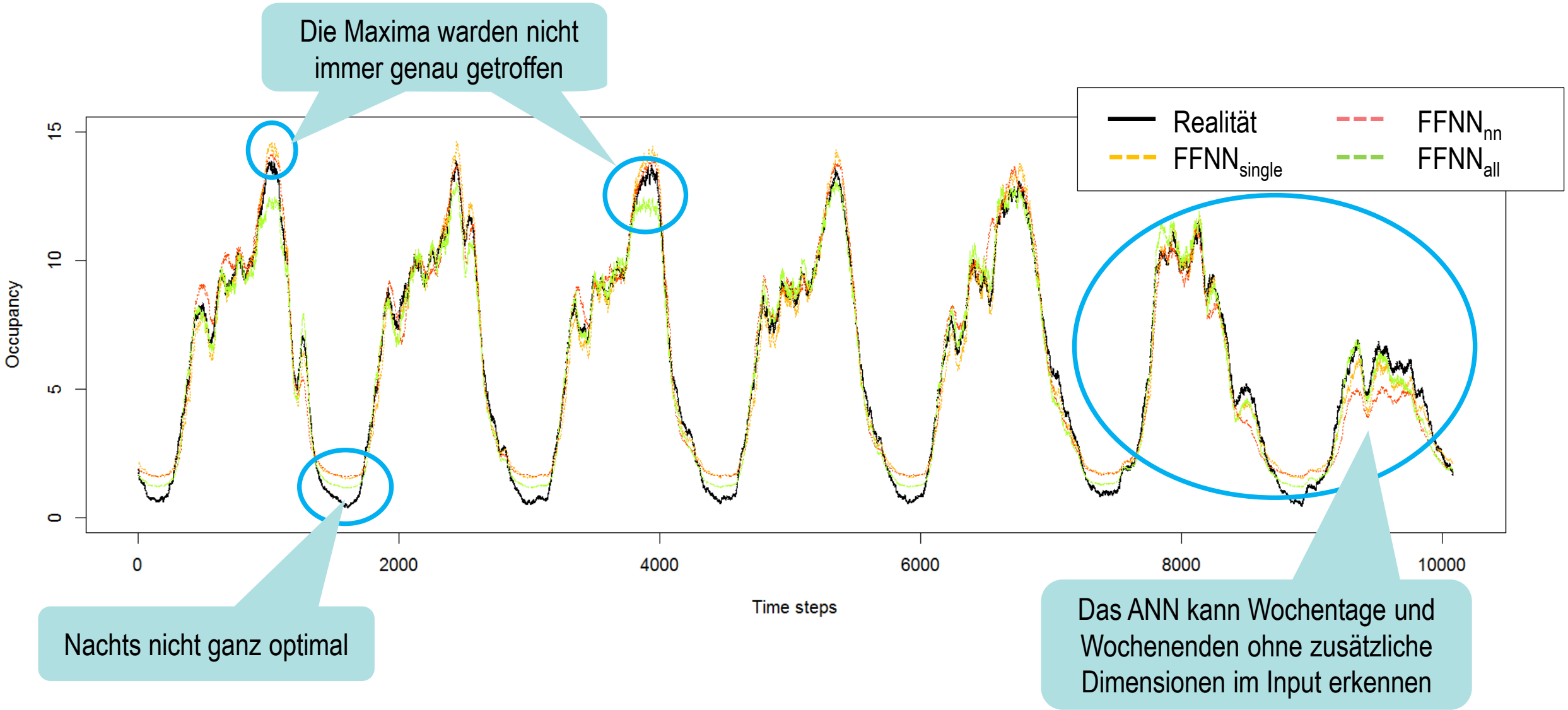
Deep Learning auf Zeitreihen

- Auswahl von Doppel-Induktionsschleifen, weil diese den Verkehr gut wiedergeben
- Erstellung einer Inputmatrix für Zeitreihen aller Sensoren →Inputvektoren für die Neuronen
- Das Target ist der Wert am Sensor in 5 / 10 / 15 / 30 / 45 Minuten
- Hyperparameter-Einstellungen?
  - Hauptsächlich Vorlagen aus Online-Tutorials





Results: One exemplary week of August



Lessons learned

- Kenne dein **Target** (beim überwachten Lernen)! Was willst du wirklich extrahieren? **Klassifikation**? **Prognose**?
  - Verkehr in Dresden ist recht regelmäßig, d.h. simple Prognosen reichen aus
  - Interessantere Frage: Wann, wo und wie entsteht Stau?

- Was ist nötig, um einen Datensatz für ein ANN aufzubereiten? (Feature Engineering, Labeling)
  - Wie hoch ist die **Qualität der Aufbereitung**? Gibt es einen **Standard**?
  - Wie gut **generalisiert** das Modell am Ende? Funktioniert es nur auf euren Daten? (**Overfitting**)

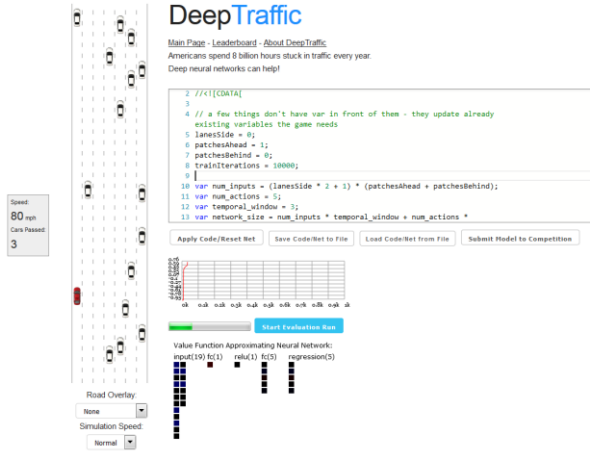
- Gibt es genügend **Varianz** in den Testdaten?
  - Ein Stauszenario bei 1000 Fällen ist zu gering („Power Law“) für einen robusten Algorithmus
  - Wo bekommt man **mehr Daten** her?

Wo anfangen?

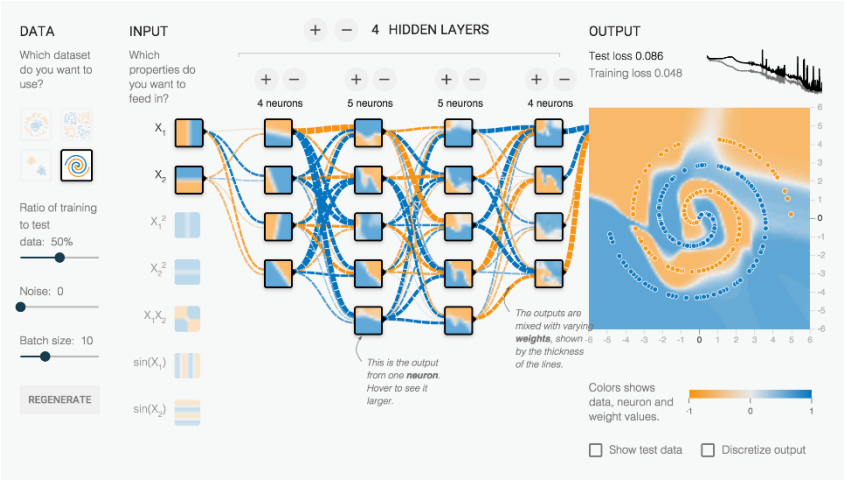
- In den letzten zwei Jahren wurden etliche **Open Source Frameworks** veröffentlicht, die verschiedene Implementierungen von ANN bereitstellen (siehe nächste Folie)

- Die Bausteine zusammen zu stecken ist meist sehr einfach. Die Aufbereitung der Daten und die Konfiguration kostet die meiste Zeit

- **Empfehlungen**:
  - Sehr simpel: Make your own neural network – Tariq Rashid (2016)
  - Profis: Deep Learning – Goodfellow, Bengio, Courville (2016)
  - Stanford Kurs. http://deeplearning.stanford.edu/tutorial/
  - TensorFlow Playground: http://playground.tensorflow.org



http://selfdrivingcars.mit.edu/deeptrafficjs/



https://blogs.scientificamerican.com/sa-visual/unveiling-the-hidden-layers-of-deep-learning/

Caffe

- C-Bibliothek von Berkeley Universität seit 2013
- Starker Fokus auf **Computer Vision** (Image Recognition)
- Hat sich in Produktivsystemen bewährt
- Komplizierte Installation, dünne Dokumentation
- Interface: C, C++, Python, MATLAB, CLI
- Plattform: Ubuntu, MacOS, Windows (experimental)

theano

- Python-Bibliothek von der University of Montreal 2007
- Paradigma symbolischer Programmierung
  - Keine schrittweise Ausführung des Skripts, sondern alles auf einmal
  - Code wird in Ausführungsgraph übersetzt, optimiert und nach C++ kompiliert
  - Sehr effektiv für numerische Berechnungen (elementar bei Deep Learning)
- Eher Low-Level für wissenschaftliche Zwecke
- Vereinfachung durch APIs wie Keras, Lasagne oder Blocks
- Seit dem Release von Tensor Flow immer weniger gefragt
- Plattform: Linux, Mac OS X, Windows

Deep Learning Software

- C++/Python-Bibliothek für ANNs von Google (OS seit 2015)
- Relativ Low-Level. Vereinfachung durch Keras API.
- Ebenfalls Paradigma symbolischer Programmierung
- TensorBoard: Schicke GUI zum Überwachen der Prozesse
- Dokumentation ist ausbaufähig aber es gibt etliche Tutorials
- Sehr aktive Entwicklercommunity = kurze Release-Zyklen
- Plattform: Linux, Mac OS X, Android, iOS, Windows

Deep Learning Software

- Lua-Bibliothek von NYU 2002 (Mitarbeit durch Facebook, Twitter, NVIDIA)
- Auch low-level aber etwas einsteigerfreundlicher als Theano oder TensorFlow, da imperativ statt deklarativ
- Schneller und etwas flexibler als seine Konkurrenten, aber weniger Funktionen
- Wer Lua nicht lernen will, kann **PyTorch** (Facebook, 2017) verwenden
- Plattform: Linux, Android, Mac OS X, iOS and Windows

Deep Learning Software

- Java-Bibliothek von Skymind 2014
- Zielt eher auf Industrie weniger auf Wissenschaft ab
- Fokus eher auf TextMining
- Integrierbar mit Apache Kafka und Apache Spark
- Sehr gute Dokumentation und Tutorials

Deep Learning Software

- MxNet (Amazon AWS)
- DSSTN - Amazon's Deep Scalable Sparse Tensor Network Engine
- Microsoft Conginitive Services (CNTK)
- Chainer
- NVIDIA DIGITS
- Paddle (Baidu)
- BigDL: Distributed Deep Learning Library for Apache Spark (Intel)

Fazit

- Deep Learning liefert **erstaunliche gute Resultate** bei der Extraktion von Informationen aus **Fuzzy Data** (Bilder, Sound, Sprache, Text, Punktwolke? etc.) – wenn man **VIELE** dieser Daten hat, sollte man definitiv erwägen, es einzusetzen

- Grenzen der **Vorstellungskraft** sind schnell erreicht, wenn man kein Mathematiker oder erfahrener Data Scientist ist. Möchte man dass? Gefahr, dass man sich zu sehr auf das ANN verlässt. ANNs können ausgetrickst werden.

- **Kein Allheilmittel**. Andere „klassischere“ Methoden können auch gut funktionieren, bzw. sind bei wenig Daten sogar besser geeignet. In den meisten Fällen wäre Deep Learning wohl momentan noch zu übertrieben

- Deep Learning läuft am besten auf **Grafikkarten** (GPUs). Komplexe Netze müssen lange trainiert werden und brauchen **sehr leistungsfähige Maschinen**.

Danke fürs Zuhören!

Gibt es Fragen?

Felix Kunde

fkunde[at]beuth-hochschule.de