

# Integrationsseminar Data Science

## Thema: Routing Lernen mit Stochastic Gradient Descent

---

Prof. Dr. Sebastian Ritterbusch

Studiengang  
Wirtschaftsinformatik

Studienakademie  
Mannheim

# Motivation

Optimierte Routenfindung ist traditionelles Gebiet der künstlichen Intelligenz

- ▶ Navigationssystem und Routenfinder sind alltägliche Hilfsmittel
- ▶ Routinglösungen abseits Fahrzeugrouting sind kaum vorhanden

# Motivation

Optimierte Routenfindung ist traditionelles Gebiet der künstlichen Intelligenz

- ▶ Navigationssystem und Routenfinder sind alltägliche Hilfsmittel
- ▶ Routinglösungen abseits Fahrzeugrouting sind kaum vorhanden

Völlig unterschiedliche Anforderungen im Routing z.B. für:

- ▶ Individuelle E-Mobility (Roller, E-Bike)
- ▶ Unerfahrene und besonders gefährdete Individuen (Schulwege)
- ▶ Menschen mit eingeschränkten Sinnen (Blinde)
- ▶ Bewegungseingeschränkte Menschen (Alter, Rollstuhl)

# Motivation

Optimierte Routenfindung ist traditionelles Gebiet der künstlichen Intelligenz

- ▶ Navigationssystem und Routenfinder sind alltägliche Hilfsmittel
- ▶ Routinglösungen abseits Fahrzeugrouting sind kaum vorhanden

Völlig unterschiedliche Anforderungen im Routing z.B. für:

- ▶ Individuelle E-Mobility (Roller, E-Bike)
- ▶ Unerfahrene und besonders gefährdete Individuen (Schulwege)
- ▶ Menschen mit eingeschränkten Sinnen (Blinde)
- ▶ Bewegungseingeschränkte Menschen (Alter, Rollstuhl)

Bisheriger und neuer Ansatz:

- ▶ Manuell definierte Regeln auf vermuteten Anforderungen und vermuteten Daten
- ▶ Automatisierte Regelfindung und Optimierung durch Stochastic Gradient Descent

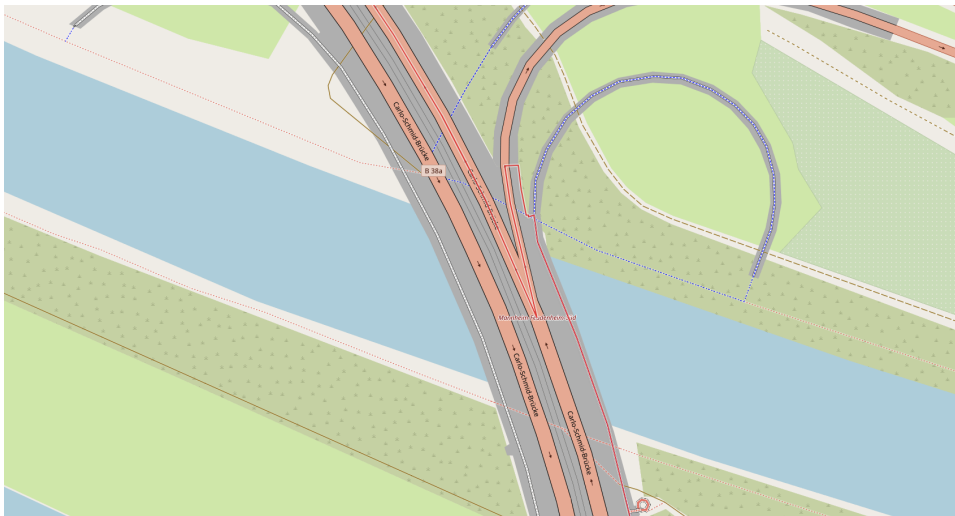
# Naives A\*-Routing auf der OpenStreetMap

Coblitz-Allee nach Käfertal:



## Naives A\*-Routing auf der OpenStreetMap

Coblitz-Allee nach Käfertal:



# Naives A\*-Routing auf der OpenStreetMap

Coblitz-Allee nach Käfertal:

Suchen

Wo ist das?

Los

Weg: Carlo-Schmid-Brücke (692610195)

Version #3

added bridge details

Bearbeitet vor ca. einem Jahr von [hofoen](#) ·  
Änderungssatz #73509690

Tags

layer	2
man_made	bridge
name	Carlo-Schmid-Brücke
wikidata	Q1041791
wikipedia	de:Carlo-Schmid-Brücke

# Daten in der OpenStreetMap

Die OpenStreetMap kennt drei physische Objekttypen:

- ▶ Node mit numerischer ID
  - ▶ Geographische Position (Längen- & Breitengrad)
  - ▶ Punkt-Informationen (Name, Hindernis, ...)
- ▶ Way mit numerischer ID
  - ▶ Liste von Punkten
  - ▶ Weg-Informationen (Name, Typ, ...)
- ▶ Relation mit numerischer ID
  - ▶ Liste von Punkten und Wegen
  - ▶ Relations-Informationen (Name, ...)



# Datenqualität in der OpenStreetMap und Folgen

Es gibt Datenstandards. Viele davon. Regional unterschiedlich. Zeitlich veränderlich.

- ▶ Von „Weg“ ohne alles,
- ▶ bis zu Stufenhöhen in Millimetergenauigkeit
- ▶ und 3D-Innenraumkarten mit Wikipedia-Links

# Datenqualität in der OpenStreetMap und Folgen

Es gibt Datenstandards. Viele davon. Regional unterschiedlich. Zeitlich veränderlich.

- ▶ Von „Weg“ ohne alles,
- ▶ bis zu Stufenhöhen in Millimetergenauigkeit
- ▶ und 3D-Innenraumkarten mit Wikipedia-Links

Interpretation von Wegen und Punkten ist regional und zeitlich unterschiedlich.

- ▶ Regeln müssen kontinuierlich der Datenlage angepasst werden.
- ▶ Ohne maschinelles Lernen ist daher kein angepasstes Routing möglich.
- ▶ Lerndaten können sogar ohne Programmiererfahrung erfasst werden.

## Satz

**Boolesche Normalformen.** Sei  $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$  eine boolesche Funktion, und die Notation  $a^{(0)} = \neg a$  und  $a^{(1)} = a$ . Dann hat  $f$  zwei bis auf Kommutativität eindeutige Darstellungen als **Disjunktive Normalform (DNF)** als boolescher Ausdruck

$$f(x_1, \dots, x_n) = \bigvee_{(a_1, \dots, a_n) \in f^{-1}(\{1\})} \left( x_1^{(a_1)} \wedge \dots \wedge x_n^{(a_n)} \right) .$$

# Bewertungsansatz

## Satz

**Boolesche Normalformen.** Sei  $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$  eine boolesche Funktion, und die Notation  $a^{(0)} = \neg a$  und  $a^{(1)} = a$ . Dann hat  $f$  zwei bis auf Kommutativität eindeutige Darstellungen als **Disjunktive Normalform (DNF)** als boolescher Ausdruck

$$f(x_1, \dots, x_n) = \bigvee_{(a_1, \dots, a_n) \in f^{-1}(\{1\})} \left( x_1^{(a_1)} \wedge \dots \wedge x_n^{(a_n)} \right).$$

```
exampleRules = { "W:sidewalk==no" : ( 10, 0),
                 "W:foot==no" : (100, 0),
                 "N:foot==no" : ( 0,1000),
                 "W:sidewalk==separate" : (100, 0),
                 "W:highway==trunk" : ( 20, 0),
                 "N:crossing==no" : ( 0, 100),
                 "W:lanes==2" : ( 10, 0),
                 "W:lanes==2 && W:sidewalk==both" : (-10, 0),
                 "W:highway==primary" : ( 30, 0),
                 "W:highway==primary && W:sidewalk==both" : (-27, 0),
                 "W:highway==secondary" : ( 20, 0),
                 "W:highway==secondary && W:sidewalk==both" : (-18, 0),
                 "W:highway==tertiary" : ( 10, 0),
                 "W:highway==tertiary && W:sidewalk==both" : ( -9, 0) }
```

# Stochastic Gradient Descent

## Stochastic Gradient Descent Verfahren im Maschinelles Lernen

- ▶ Ein neuronales Netz bildet Lerndaten auf Ausgaben ab.
- ▶ Ergibt einen Unterschied des Resultats zu erwünschten Ausgaben.
- ▶ Unterschied wird zurückgerechnet auf Knoten.

# Stochastic Gradient Descent

## Stochastic Gradient Descent Verfahren im Maschinelles Lernen

- ▶ Ein neuronales Netz bildet Lerndaten auf Ausgaben ab.
- ▶ Ergibt einen Unterschied des Resultats zu erwünschten Ausgaben.
- ▶ Unterschied wird zurückgerechnet auf Knoten.

## Lerndaten im lernenden Routing

- ▶ Routing von A nach B
- ▶ Erwünschte Via-Punkte als Anforderungen
- ▶ Unterschied der Routen „direkt“ vs. über „via-Punkte“ liefert Regelwirkung
- ▶ Unterschied wird zurückgerechnet auf Regeln

# Stochastic Gradient Descent

## Stochastic Gradient Descent Verfahren im Maschinelles Lernen

- ▶ Ein neuronales Netz bildet Lerndaten auf Ausgaben ab.
- ▶ Ergibt einen Unterschied des Resultats zu erwünschten Ausgaben.
- ▶ Unterschied wird zurückgerechnet auf Knoten.

## Lerndaten im lernenden Routing

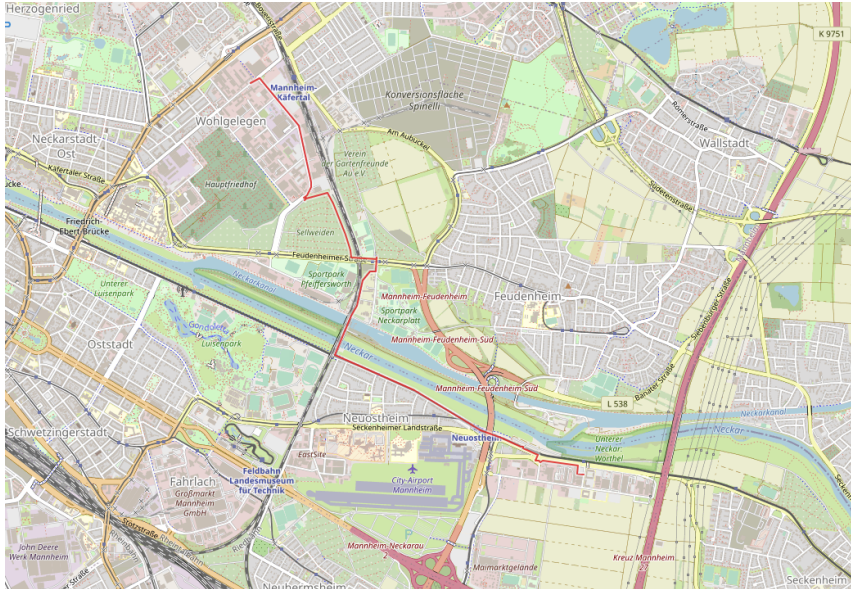
- ▶ Routing von A nach B
- ▶ Erwünschte Via-Punkte als Anforderungen
- ▶ Unterschied der Routen „direkt“ vs. über „via-Punkte“ liefert Regelwirkung
- ▶ Unterschied wird zurückgerechnet auf Regeln

## Regelmodifikation

- ▶ Unterschied der Routen liefert auch mögliche neue Regeln
- ▶ En-passant Änderung der Regelmatrix

# DNF-Regel A\*-Routing auf der OpenStreetMap

Coblitz-Allee nach Käfertal:





# Gruppenaufgabe

Gegeben:

- ▶ OSM-Daten weltweit
- ▶ Beispiel-Routen und Zielpunkt z.B. für Fußgänger um DHBW Mannheim
- ▶ A\*-Router mit DNF-Regelbewertung in Python

# Gruppenaufgabe

Gegeben:

- ▶ OSM-Daten weltweit
- ▶ Beispiel-Routen und Zielpunkt z.B. für Fußgänger um DHBW Mannheim
- ▶ A\*-Router mit DNF-Regelbewertung in Python

Aufgabe:

- ▶ Stochastical Gradient Descent auf festem Regelwerk
- ▶ Dynamische Regelerweiterung und -elimination
- ▶ Ergebnisse evaluieren

Literatur: S. Ritterbusch, H. Kucharek: Robust and Incremental Pedestrian Path Network Generation on OpenStreetMap for Safe Route Finding. Springer, Cham, 2018.