

# Einsatz von Neuronalen Netzwerken und Clusteringverfahren auf Geodatensätze

Lehrstuhl für Geoinformatik

Robin Bially

[HTTPS://GITHUB.COM/ROBINBIA/PROJEKTARBEIT-GEOINFORMATIK.GIT](https://github.com/ROBINBIA/PROJEKTARBEIT-GEOINFORMATIK.GIT)

Projektarbeit unter der Betreuung von PD Dr. Dr.-Ing. Wilfried Linder von 11.2017 - 11.2018 als Vorbereitung der sich anschließenden Masterarbeit.

*Fertigstellung, November 2018*



# Inhaltsverzeichnis

<b>1</b>	<b>Motivation</b>	<b>5</b>
1.1	Ieeres Kapitel	5
<b>2</b>	<b>Geodaten und Geoinformation</b>	<b>7</b>
2.1	Definition und Gestalt von Geodaten	7
2.2	Geografische Koordinaten	8
2.3	Qualitätsmerkmale	8
2.4	Georeferenzierung	11
2.4.1	Definition	11
2.4.2	Adresskodierung	11
2.4.3	Geotagging	11
2.4.4	Kartenkalibrierung	11
2.4.5	Rektifizierung	11
2.4.6	Bestimmung einer Transformationsvorschrift	11
2.5	Geoinformationssysteme	12
2.5.1	Geoobjekte	12
2.5.2	Modellierung von Geoobjekten	12
2.5.3	Rastermodell	13
2.5.4	Vektormodell	13
2.6	Beispiele von Raster und Vektordaten	15
2.7	Algorithmen in der Geoinformatik	16
2.8	Verschiedene Arten und ihre Anwendungszwecke	16

<b>3</b>	<b>Deep Learning .....</b>	<b>17</b>
<b>3.1</b>	<b>Was ist Machine Learning?</b>	<b>17</b>
<b>3.2</b>	<b>Motivation und Anwendungsgebiete</b>	<b>17</b>
3.2.1	Linear Classifier .....	19
3.2.2	Regularisierer .....	20
3.2.3	Gradientenabstieg .....	21
3.2.4	Multi-Layer Neural Network .....	21
3.2.5	Trainieren eines Netzwerks mittels Backpropagation .....	21
3.2.6	Parameterupdates <sup>2</sup> .....	23
3.2.7	Convolutional Neural Network .....	24
3.2.8	Recurrent Neural Network .....	25
3.2.9	Long Short Term Memory Netzwerke .....	26
<b>3.3</b>	<b>Machine Learning in Geowissenschaften</b>	<b>28</b>
3.3.1	Herausforderungen und Chancen von Machine Learning (Kar+17) .....	28
3.3.2	Einsatz in der Praxis - Beispiele .....	32
<b>3.4</b>	<b>Tensorflow</b>	<b>32</b>
<b>3.5</b>	<b>Erstes eigenes CNN</b>	<b>32</b>
<b>4</b>	<b>Clusteringverfahren .....</b>	<b>33</b>
<b>4.1</b>	<b>Ziel</b>	<b>33</b>
<b>4.2</b>	<b>Eindeutiges Clustering</b>	<b>33</b>
4.2.1	Partitionierende Verfahren und ihre Nachteile .....	33
4.2.2	Dichtebasierte Verfahren .....	33
4.2.3	Hierarchische Verfahren .....	33
4.2.4	Outlier Detection .....	33
<b>4.3</b>	<b>Fuzzy-Clustering</b>	<b>33</b>
4.3.1	Probabilistisches und Possibilistisches Clustering .....	33
<b>4.4</b>	<b>CVI</b>	<b>33</b>
4.4.1	NPC .....	33
4.4.2	FHV .....	33
4.4.3	Otsu-Binarisierung .....	33
4.4.4	VAT-Algorithmus .....	33
<b>4.5</b>	<b>Clustering auf unvollständigen Daten</b>	<b>33</b>
<b>4.6</b>	<b>Einsatz in der Praxis und Software-Tools</b>	<b>33</b>
<b>5</b>	<b>Fazit und Ausblick .....</b>	<b>35</b>
<b>5.1</b>	<b>Ausblick - Mein Thema für die Masterarbeit</b>	<b>35</b>

---

<sup>2</sup><http://ruder.io/optimizing-gradient-descent/>



# 1. Motivation

## 1.1 leeres Kapitel

blabla





## 2. Geodaten und Geoinformation

### 2.1 Definition und Gestalt von Geodaten

Geodaten sind digitale Informationen, welche Sachdaten mit Geometriedaten<sup>1</sup> (und Chronometriedaten) vereinen , z.B. {Luftdruck 1 bar, Ort Düsseldorf, Datum 26.11.2017}. Die räumliche Information kann in unterschiedlichen Formen vorliegen, z.B. symbolisch als Ortsname oder Postleitzahl, aber auch als mathematisch atomare Referenz auf Positionen der Erde mittels Koordinaten. Diese können in unterschiedlichster Dimensionalität vorliegen<sup>2</sup>

- Ein Objekt ohne bestimmte Länge (0D)
- Ein Liniensegment (1D)
- Gauß-Krüger oder geografische Koordinaten mit Bezug auf die Oberfläche der Erde ohne Berücksichtigung von Höhenunterschieden (2D)
- 2D-Koordinaten mit einer zusätzlichen Sachinformation für die Höhe über dem Geoiden (2.5D).
- Kugelkoordinaten mit Bezug auf jeden Punkt im Volumen der Erde als Geoid oder Rotationsellipsoid (3D)
- Zusätzlich zu den 3 Koordinaten im Raum wird eine vierte Information mitgeführt, die sich aus dem zeitlichen Ablauf ergibt (4D)

---

<sup>1</sup><https://www.hdm-stuttgart.de/riekert/lehre/gis.pdf>

<sup>2</sup><http://www.mathematik.uni-ulm.de/sai/ws04/biosem/GIS.pdf>

## 2.2 Geografische Koordinaten

Ein geeignetes und weit verbreitetes Koordinatensystem zur verzerrungsfreien Darstellung sind die Geografischen Koordinaten.

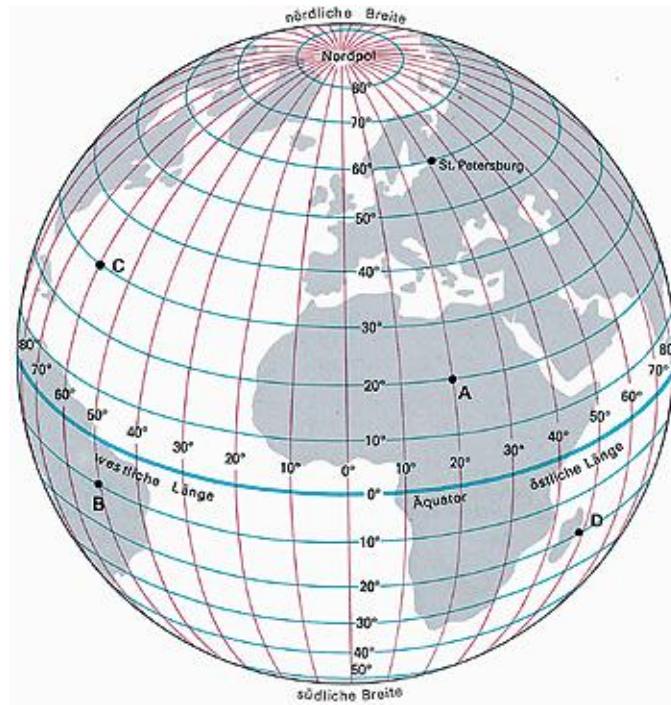


Abbildung 2.1: Das Gradnetz der Erde

Beschrieben wird ein Punkt auf der Erde durch gedachte Kreise um den Globus, welche senkrecht zueinander stehen. Insgesamt existieren 180 Breitenkreise (Richtung Ost-West) und 360 Längenkreise (Richtung Nord-Süd). Die Abweichung von den beiden Referenzkreisen Äquator und Nullmeridian wird in Grad östlicher/westlicher Länge und nördlicher/südlicher Breite angegeben. Als Äquator ( $0^\circ$  nördliche/südliche Breite) wird der Breitenkreis bezeichnet, auf welchem die Erdachse senkrecht steht. Der Nullmeridian ( $0^\circ$  westliche/östliche Länge) ist der Längenkreis, welcher durch die britische Stadt Greenwich verläuft. Weitere wichtige Koordinatensysteme sind die Gauß-Krüger und UTM-Koordinaten. Die Vorteile dieser Systeme ist, dass sich eine geografische Position direkt ablesen lässt. Geografische Koordinaten erschweren dies bedingt durch die sich verändernden Abstände zwischen den Längenkreisen in zunehmender Nord- oder Südrichtung.

## 2.3 Qualitätsmerkmale

Ein wichtiger Forschungszweig ist die automatische Beurteilung von Qualitätsmerkmalen von Geodaten hinsichtlich einer bestimmten Fragestellung. Ein geeignetes Maß ist die gewichtete Summe verschiedener Datenmerkmale, welche in der aktuellen ISO-Norm *ISO 19157:2013*<sup>3</sup> spezifiziert sind.

<sup>3</sup><https://www.iso.org/standard/32575.html>

Die Beurteilung der Datenqualität ist ebenfalls in sämtlichen Verarbeitungsschritten des KDD-Prozessmodells[FPS96] erforderlich. Das KDD-(Knowledge Discovery in Databases-)Prozessmodell beschreibt, wie aus einer unstrukturierten Datensammlung neues, relevantes Wissen extrahiert werden kann. Es umfasst folgende Schritte:

1. **Fokussieren/Selektieren** Die Daten werden zusammengetragen, in Form einer Datenbank strukturiert und irrelevante Einträge gelöscht. Relevante Daten haben das Potenzial hinsichtlich eines vorgegebenen Ziels vielversprechende Erkenntnisse zu liefern.
2. **Vorverarbeitung** Es wird überprüft, ob die Datenbank konsistent ist und ob es fehlende Einträge gibt. Die Konsistenz wird verletzt, wenn zum Beispiel einer Koordinate ein Land zugeordnet wird, welches den umgebenen Datenpunkten nicht zugeordnet wurde. Die Wahrscheinlichkeit einer Fehlzuordnung ist dann groß. Die Datenbank hat fehlende Einträge, wenn es Attribute mit nicht existierenden Werten gibt. Strategien zum Umgang mit fehlerhaften Datensätzen beim Clustering werden im Kapitel über Clusteringverfahren erläutert.
3. **Transformation** Wurde ein komplexes Objekt in die Datensammlung aufgenommen, so müssen seine Merkmale zuerst numerisch diskretisiert werden. Beispielsweise hat ein Reisfeld eine Länge und Breite, aber auch eine Vegetationsdichte und einen Erntestatus. All diese Merkmale müssen vermessen und in die Datenbank aufgenommen werden. Merkmale, welche nach der Transformation in mangelhafter Qualität vorliegen, können entfernt werden.
4. **Data Mining** Methoden zur Mustererkennung. Dazu gehören sowohl Neuronale Netze und Clusteringverfahren, als auch Assoziationsregeln. In folgenden Kapiteln werden diese erklärt.
5. **Evaluation** Ergebnisse der Mustererkennung werden statistisch überprüft und die Nützlichkeit vom Nutzer bewertet.

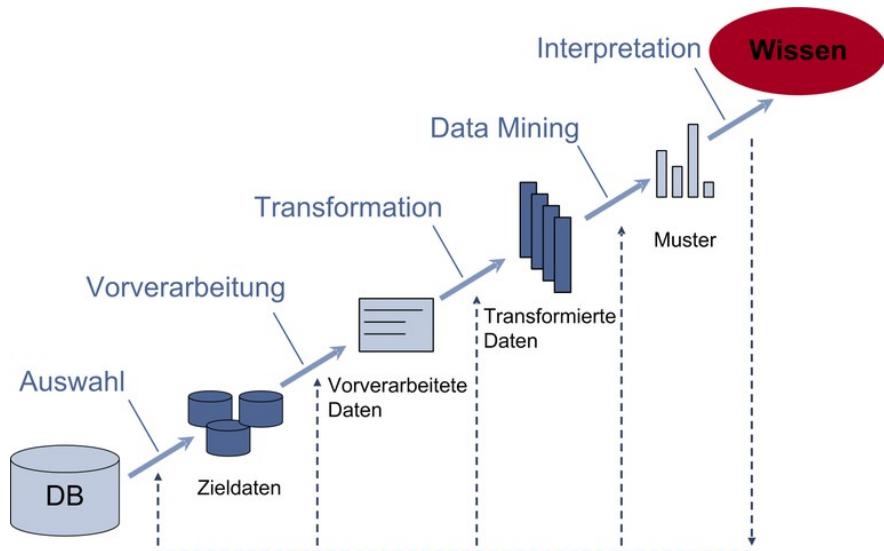


Abbildung 2.2: Das KDD-Prozessmodell <sup>4</sup>

<sup>4</sup><http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/daten-wissen/Business-Intelligence/Analytische-Informationssysteme--Methoden-der-/Data-Mining/index.html>

Die folgende Auflistung ist eine informelle Beschreibung der oben genannten ISO-Norm *ISO 19157:2013* durch Fragestellungen und Beispiele:

- **Vollständigkeit**
  - **Datenüberschuss** - Enthält der Datensatz mehr Objekte und Beziehungen als angegeben?
  - **Datenmangel** - Enthält der Datensatz weniger Objekte und Beziehungen als angegeben?
- **Logische Konsistenz**
  - **Konzeptuelle Konsistenz** - Wurde die Gestalt des Datenmodells bei Aktualisierungen nicht verändert?
  - **Wertekonsistenz** - Sind alle Werte sinnvoll?
  - **Formatkonsistenz** Passen die Daten zu angegebenen physikalischen Einheiten?
  - **Topologische Konsistenz** Bleiben topologische Beziehungen bei Änderungen des Datensatzes bestehen (Der botanische Garten befindet sich im Umkreis von 1km von der HHU)?
  - **Geometrische Konsistenz** - Ist der digitalisierte Datensatz geometrisch sinnvoll und widerspruchsfrei?
- **Positionsgenauigkeit**
  - **Äußere Genauigkeit** - Wie gut stimmen die Koordinatenwerte des Datensatzes mit den wahren Koordinaten überein?
  - **Innere Genauigkeit** - Wie gut stimmen die relativen Positionen von Objekten zueinander mit den wahren relativen Positionen überein?
  - **Rasterdatengenauigkeit** - Wie gut stimmen die Rasterdatenpositionswerte mit den wahren Werten überein?
- **Zeitliche Genauigkeit**
  - **Genauigkeit von Zeitmessungen** - Wie genau ist die Zeitangabe (minutengenau, taggenau)?
  - **Zeitliche Konsistenz** - Ist die Reihenfolge der Ereignisse korrekt?
  - **Zeitliche Gültigkeit** - Ist der Datensatz in Bezug auf das geforderte Zeitformat korrekt?
- **Thematische Genauigkeit**
  - **Richtigkeit der Klassifikation** - Stimmen Objekte, oder ihre Attribute mit den zugewiesenen Klassen überein, z. B. Zuordnung zu Fluss, statt zu Weg
  - **Richtigkeit nichtquantitativer Attribute** - Beispiel: Ist das Grundstück wirklich eine Bananenplantage?
  - **Genauigkeit quantitativer Attribute** - Beispiel: Ist die Fläche des Grundstücks korrekt?

Viele der oben genannten Punkte lassen einen subjektiven Spielraum für die Bewertung zu. Sowohl Skalierungen als auch Gewichtungen sind nicht eindeutig definiert, was einen Vergleich verschiedener Datensätze erschwert. Aus diesem Grund ist eine algorithmische Interpretation in Kombination mit Verfahren der künstlichen Intelligenz hilfreich. So ließe sich aus der Norm ein universeller und allgemeingültiger Indikator zur Bewertung der Datenqualität ermitteln.

## 2.4 Georeferenzierung

### 2.4.1 Definition

Unter dem Vorgang der Georeferenzierung versteht man die Zuweisung raumbezogener Informationen, auch Georeferenz genannt, zu einem Datensatz.

Es gibt folgende vier Arten der Georeferenzierung:

- Adresskodierung
- Geotagging
- Kartenkalibrierung
- Rektifizierung

### 2.4.2 Adresskodierung

Bei der Adresskodierung wird dem Datensatz eine Postanschrift zugewiesen und somit ein indirekter Raumbezug hergestellt. Mithilfe geokodierter Adressen lassen sich funktionale Zusammenhänge zwischen Daten, Postanschrift und Adresse herstellen und somit ressourcenschonende und schnelle Zugriffe ermöglichen.

### 2.4.3 Geotagging

Als Geotagging bezeichnet man das Einfügen eines Attributes (Geotag) inkl. Realweltkoordinate in einen raumbezogenen Datensatz wie ein Bild oder eine Website. Dies ist bei der räumlichen Einordnung der Information hilfreich.

### 2.4.4 Kartenkalibrierung

Bei der Kartenkalibrierung wird ein räumlicher Datensatz ohne Realkoordinatenbezug mithilfe einer Transformationsvorschrift im Bezug auf die Realwelt so orientiert, dass sich die Koordinaten des Bildes in Realweltkoordinaten einfach umrechnen lassen.

### 2.4.5 Rektifizierung

Bei der Rektifizierung werden geometrische Verzerrungen in räumlichen Daten entzerrt, indem jedem Datum eine Realweltkoordinate zugeordnet wird.

### 2.4.6 Bestimmung einer Transformationsvorschrit

Um eine Transformationsgleichung zu finden, werden in der Regel Passpunkte verwendet. Die Passpunkte müssen im Datensatz eindeutig zu erkennen sein. Die Koordinaten der Passpunkte im Realweltkoordinatensystem sind entweder bekannt oder werden einem Referenzdatensatz entnommen. Bei Vektordaten werden die Koordinaten abgegriffen oder interpoliert. Bei Bilddaten werden die Bildkoordinaten der Passpunkte gemessen. Die Transformation sollte unter Berücksichtigung der Abbildungsgeometrie bestimmt werden. Bei Fotos ist somit die Zentralprojektion zu berücksichtigen, bei Karten der entsprechende Kartennetzentwurf. Das automatische Finden von Gemeinsamkeiten in digitalen Bildern und die Bestimmung der Transformation wird in der Bildverarbeitung Bildregistrierung genannt. Die Registrierung von Laserscanning-Punktfolgen kann mit dem ICP-Algorithmus erfolgen.<sup>5</sup>

Mark Erweiterungen, implementierungen sinnvoll?

<sup>5</sup><https://de.wikipedia.org/wiki/Georeferenzierung>

## 2.5 Geoinformationssysteme

Ein Geoinformationssystem ist eine Software, mit welcher Geodaten erfasst, verwaltet, analysiert und ausgegeben werden können.

Man unterscheidet bei der Abfrage von Daten unter folgenden verschiedenen Typen:

- Alphanumerische Daten (Attribute als Text oder Zahlen)
- Text-Dokumente
- Multimediale Informationen, wie Videos, Audiosequenzen, Animationen
- Fotos, Scans, Satellitenbilder

Der Unterschied zu einer Datenbank ist, dass jedes Sachdatum einen expliziten Raumbezug hat, über welchen die Selektion erfolgt. In einer Datenbank erfolgen Zugriffe stattdessen über Schlüsselattribute. Eine weitere Stärke eines GIS ist die grafische Aufbereitung der Daten zur anschaulich-interaktiven Analyse.

Beispiele für solche räumlichen Analysewerkzeuge sind Routenfindung, räumliche Suche. Ein implementiertes Kartografiesystem ermöglicht zudem das markieren von Punkten und Linien, färben von Flächen und die Anzeige und Überlagerung verschiedener Ebenen.

### 2.5.1 Geoobjekte

Ein Geoobjekt ist ein tatsächlich auf der Erde vorhandenes Objekt, welches durch Geodaten eindeutig referenziert wurde. Man unterscheidet zwischen Gegenständen und Sachverhalten. Gegenstände sind konkrete, visuell wahrnehmbare Erscheinungen auf der Erdoberfläche. Sachverhalte dagegen sind nicht sofort visuell wahrnehmbar, sondern bezeichnen Beziehungen zwischen Gegenständen oder die Interaktion mit der Umwelt und Oberflächengestalt. Außerdem unterscheidet man zwischen verschiedenen Arten der Datenspeicherung:

- Flächenhafte Daten
- Linienhafte Daten
- Punkthafte Daten

Je nach Kartenmaßstab, Auflösungstyp und Speichertyp (digital/analog) werden Daten unterschiedlich repräsentiert. So wird beispielsweise ein flächenhafter quadratischer Gebäudekomplex (10\*10 Meter) auf einem Satellitenfoto mit dem Maßstab 1:10.000 nur noch als Punkt wahrgenommen. Linienhafte Daten bieten sich vor allem bei Flüssen, Straßen, Wasser-Land-Grenzen, starken Flankensteigungen usw. an.

### 2.5.2 Modellierung von Geoobjekten

Die vier informationstechnischen Dimensionen zur Modellierung von geografischen Informationssystemen sind:

- Geometrie (Ort des Objekts)
- Topologie (Lage der Objekte relativ zueinander)
- Semantik (Bedeutung des Objekts im fachspezifischen Kontext, z.B. gut-schlecht, viel-wenig, groß-klein)
- Dynamik (Änderung des Objekts im zeitlichen Verlauf)

Jedes unikate Objekt gehört zu einer Objektklasse, in welcher es nach den oben genannten vier Kriterien beschrieben und mit anderen Objekten der Klasse verglichen wird. Jedes der Objekte besitzt einen eindeutigen Schlüssel zur Identifikation. Möglichkeiten zur Klassifizierung und Clustering von Geoobjekten werden in den folgenden Kapiteln vorgestellt.

Der Ort eines Geoobjektes kann auf zwei verschiedene Arten beschrieben werden:

### 2.5.3 Rastermodell

Eine analoge topografische Karte oder Zeichnung digitalisiert und in quadratische Gitterzellen aufgeteilt, welche alle über die gleiche Semantik verfügen. Diese Semantik wird stellvertretend durch eine Matrix beschrieben, welche für jede Gitterzelle eine numerische Pixelwertinformation enthält.<sup>6</sup>

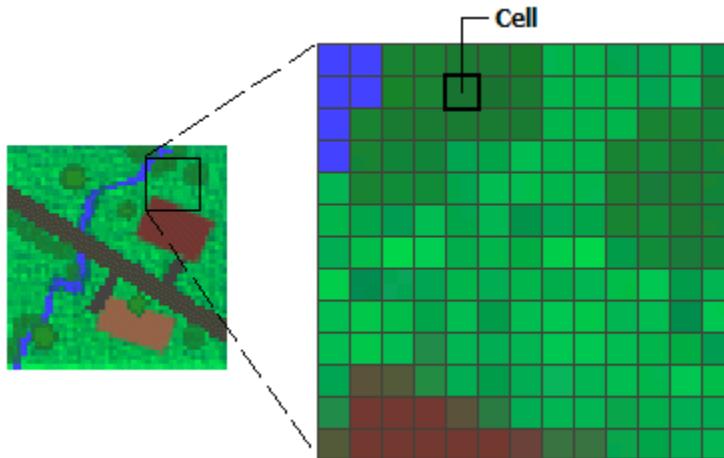


Abbildung 2.3: Rastermodell-Zoom<sup>7</sup>

Diese Pixelinformationen repräsentieren Daten wie Temperatur, Höhe, Vegetationsdichte, Landnutzung, Bodenbeschaffenheit. Rasterdaten werden in der Regel als Bilddatei gespeichert (BMP, GIF, JPEG).

Die Rastergeometrie eignet sich gut zur Beschreibung flächiger, homogener Sachverhalte. Die einfache Struktur bietet viele Vorteile, aber auch Nachteile:

#### Vorteile

- Einfache Datenstruktur
- Geeignet für räumliche und statistische Analyse
- Alles ist einheitlich speicherbar (Punkte, Linien, Polygone)
- Überlagerung von Ebenen sehr schnell und einfach

#### Nachteile

- Genauigkeitsverlust beim Scannen und Neustrukturieren
- Endliche Auflösung => räumliche Ungenauigkeit
- Pixelwerte haben keine Beziehung zueinander
- Hoher Speicheraufwand bei Hoher Auflösung, keine Kompression möglich.

### 2.5.4 Vektormodell

Im Gegensatz zu Rasterdaten werden Vektordaten bei linien- und punkthaften Informationen eingesetzt, also Informationen, die sich nicht mit homogener Eigenschaft über die gesamte Karte verteilen. Man nennt solche Informationen auch Features. Beispiele hierfür sind Straßen, Staatsgrenzen, Gewässergrenzen, Höhenlinien, Flüsse, Bäume.

Eine Punkthafte Vektorinformation wird auch als Vertex bezeichnet. Dieser beschreibt eine

<sup>6</sup><https://de.wikipedia.org/wiki/Geoobjekt>

<sup>7</sup><http://desktop.arcgis.com/de/arcmap/10.3/manage-data/raster-and-images/what-is-raster-data.htm>

Raumlage durch Angabe einer (x,y,z)-Koordinate und ein dazugehöriges Attribut, welches die Art des Punktes beschreibt, z.B. Baum oder Laterne:

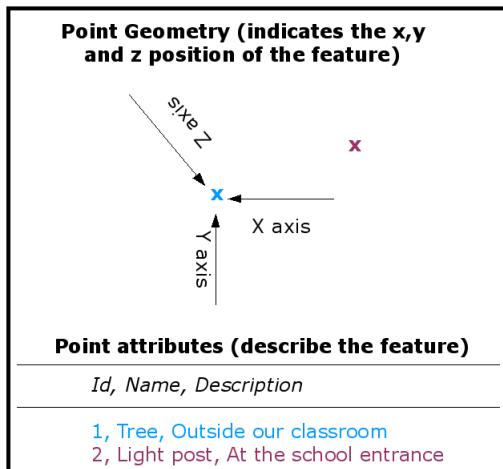


Abbildung 2.4: Punkt-Feature<sup>8</sup>

Punkteverläufe wie Straßen werden durch sogenannte Polylinien beschrieben. Diese bestehen aus mehreren miteinander verbundenen Vertices. Im Kreis laufende Polylinien bezeichnet man auch als Polygone:

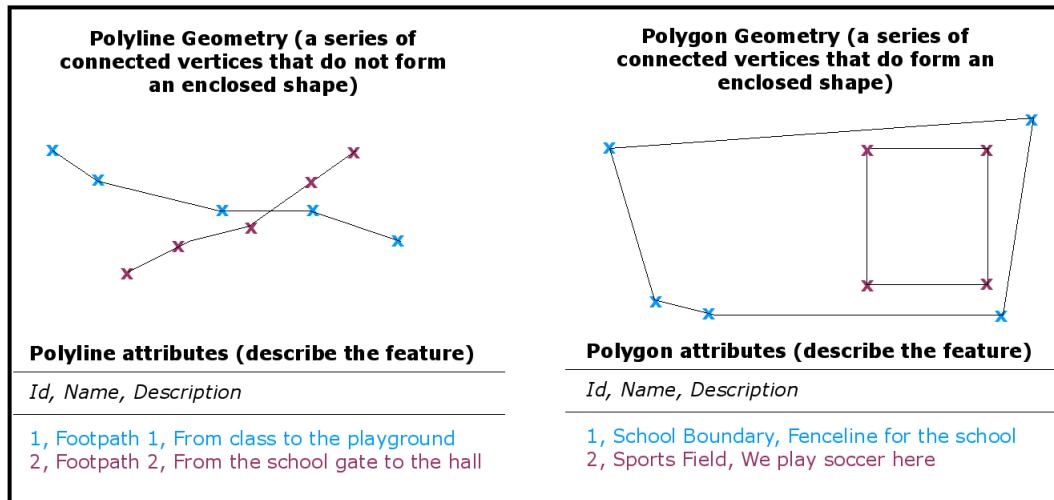


Abbildung 2.5: Polylinien und Polygone<sup>9</sup>

Wie auch bei Rasterdaten gibt es bei Vektordaten nicht nur Vorteile, sondern auch Nachteile.<sup>10</sup>

#### Vorteile:

- Unendliche Linienauflösung und sehr hohe Genauigkeit

<sup>8</sup>[https://docs.qgis.org/2.8/de/docs/gentle\\_gis\\_introduction/vector\\_data.html](https://docs.qgis.org/2.8/de/docs/gentle_gis_introduction/vector_data.html)

<sup>9</sup>[https://docs.qgis.org/2.8/de/docs/gentle\\_gis\\_introduction/vector\\_data.html](https://docs.qgis.org/2.8/de/docs/gentle_gis_introduction/vector_data.html)

<sup>10</sup><http://romanharcke.de/geoinformationssysteme-geodaten-kapitel-4/>

- Beschreibung von mehreren einzigartigen Features in nur einer Ebene möglich
- Geringer Speicherbedarf
- Einfache Erzeugung von Topologie (Knoten, Kanten, Flächen)
- Gute Performance
- Ermöglicht Attributierung und Objektdefinitionen

#### Nachteile:

- Flächenhafte Informationen können nicht gespeichert werden
- Durch Scannen können diese Daten nicht erzeugt werden. Es bedarf hier einer Raster-Vektorwandlung (Hoher Erfassungsaufwand)
- Hoher Rechenaufwand bei Verschneidungen

## 2.6 Beispiele von Raster und Vektordaten

Geodaten müssen heutzutage nicht mehr selbständig erstellt werden. Es gibt eine Vielzahl an staatlichen und privaten Institutionen, welche Ihre Daten kostenlos bereitstellen. So lassen sich zahlreiche Inhalte im ESRI Shapefile Vektordateiformat finden, welches als Quasi-Standard für Desktop-GIS gilt.<sup>11</sup> Der Datensatz *Natural Earth*<sup>12</sup> ist eine Abbildung der Erde im Maßstab 1:10 Millionen. Er ist sowohl als SHP-Vektordatei als auch als Tiff-Rasterbild verfügbar.

Ein ESRI Shapefile besteht aus mindestens drei Dateien zur Speicherung der Geometriedaten, Sachdaten und der Geometrieindizierung zur Verknüpfung von Geometrie und Sachdaten (.shp, .dbf, .shx). Die Geometrie eines Shapefile definiert sich aus nur 4 verschiedenen Formdatenstrukturen: Punkte, Linien, Flächen (Polygone) und Multipunkte.<sup>13</sup>

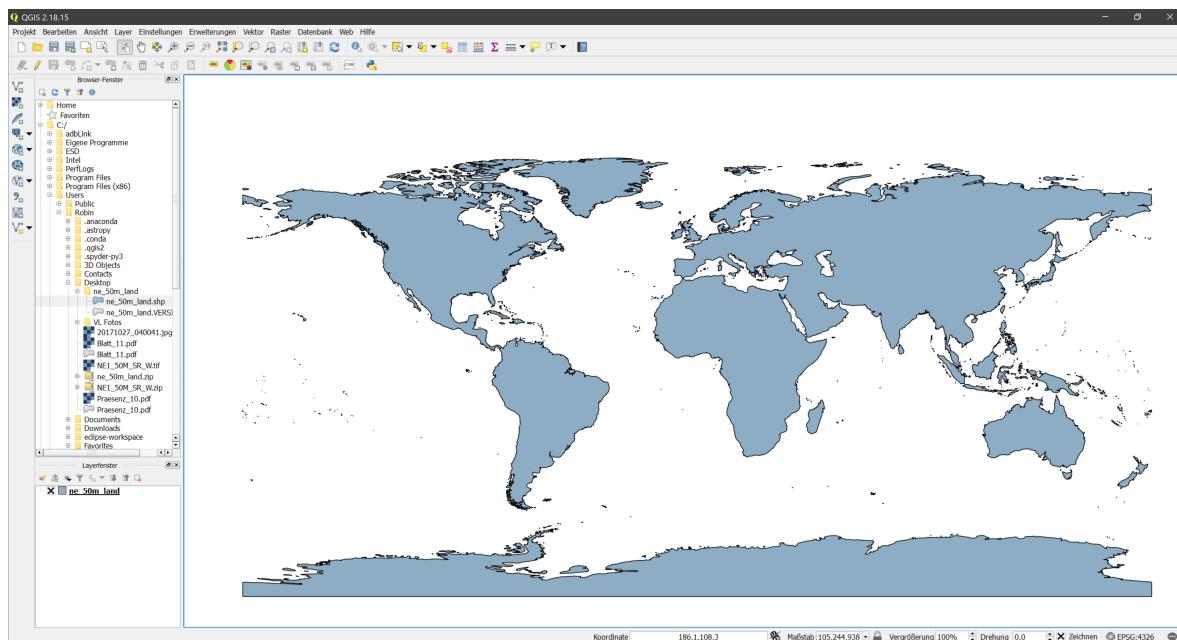


Abbildung 2.6: .shp-Geometriedatei in dem Geoinformationssystem QGIS dargestellt<sup>14</sup>

<sup>11</sup><https://de.wikipedia.org/wiki/Shapefile>

<sup>12</sup><http://www.naturalearthdata.com/>

<sup>13</sup><http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

Leider eignen sich Vektordaten nicht zur Klassifikation von Features mit Hilfe von Deep Learning wie z.B. Convolutional Neural Networks (CNN), sondern stellen viel mehr das Ergebnis einer Rasterbildanalyse dar. Aus diesem Grund beziehen sich folgende Kapitel im Kontext von Geodaten immer auf Rasterdaten und Bildausschnitte.

Das dem Datensatz zugehörige farbige Rasterbild inklusive Schummerung (räumliche Schattierung), Wasser und Flüssen sieht so aus:

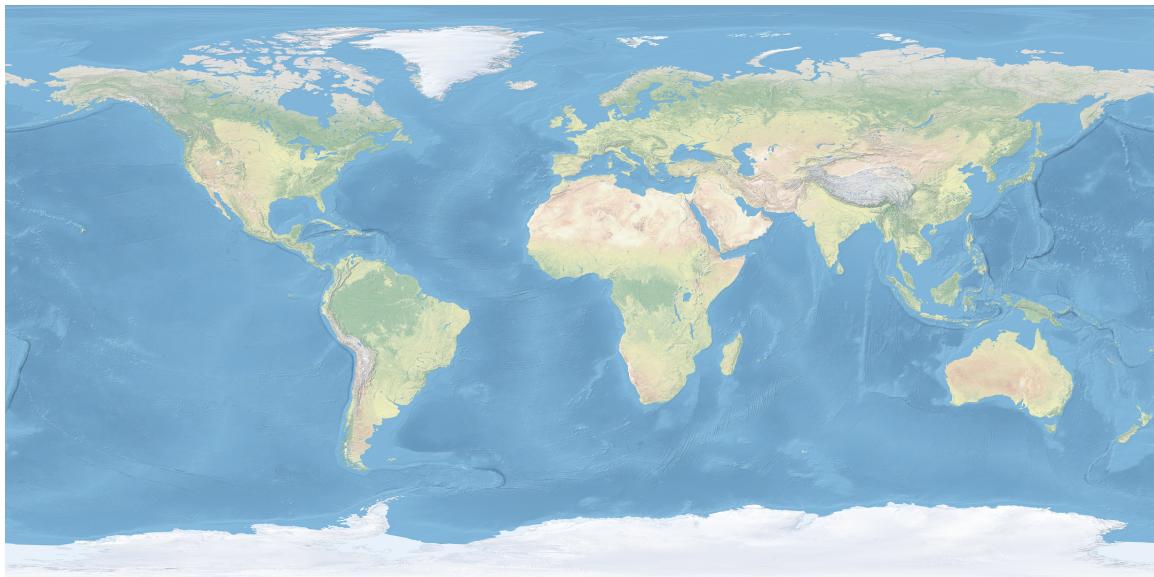


Abbildung 2.7: Rasterbild des Natural-Earth-Datensatzes<sup>15</sup>

## 2.7 Algorithmen in der Geoinformatik

## 2.8 Verschiedene Arten und ihre Anwendungszwecke

---

<sup>14</sup><https://www.qgis.org/de/site/>

<sup>15</sup><http://www.naturalearthdata.com/downloads/10m-raster-data/10m-natural-earth-1/>



## 3. Deep Learning

### 3.1 Was ist Machine Learning?

#### Definition

Machine Learning ist eine Unterdisziplin der künstlichen Intelligenz und basiert auf der Idee, biologische Denkprozesse, wie sie in Gehirnen ablaufen, nachzuahmen[Lar+15].

*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.[Mit97]*

Ein Computerprogramm lernt also genau dann dazu, wenn es sich hinsichtlich seiner Performance in bestimmten Aufgabengebieten mithilfe von Erfahrung selbstständig verbessert.

### 3.2 Motivation und Anwendungsbereiche

Ziel von Machine Learning in den Geowissenschaften ist es, Muster in Geodaten zu erkennen, Vorhersagen zu machen und Phänomene besser erkennen und verstehen zu können.

Eine Stadt ist ein Komplexes System, das aus vielen kleineren interagierenden Subsystemen besteht. Diese werden durch Faktoren wie Politik, Bevölkerungswachstum, Verkehrsinfrastruktur und den Arbeitsmarkt beeinflusst. Um zu verstehen, welche Kräfte strukturelle Änderungen von Städten vorantreiben, werden sowohl Satellitenbilder als auch nutzerbezogene Positionsdaten aus sozialen Netzwerken wie Facebook und Twitter und Attributierte Markierungen auf Geoinformationssystemen wie OpenStreetMap<sup>1</sup> verwendet, um Langzeitvorhersagen zur erstellen. Außerdem helfen diese Modelle und Simulationen dabei, die Mechanismen der urbanen Evolution zu erforschen und Städteplanung zu optimieren.

Im Folgenden eine Auflistung verschiedener Probleme, für deren Lösung sich die Anwendung eines Neuronalen Netzwerks eignet:

---

<sup>1</sup><https://www.openstreetmap.org>

- Klassifizierung - Was ist auf einem Bild zu sehen?
- Lokalisation - Wo ist das Objekt auf dem Bild?
- Segmentierung - Klassifizierung jedes Pixels
- Lineare Regression - Lässt sich ein funktionaler Zusammenhang zwischen den Daten des Datensatzes finden, welcher eine Vorhersage zum weiteren Verlauf der Daten ermöglicht?
- Clustering - Wie lassen sich Daten vergleichen und in Gruppierungen bei gewisser Ähnlichkeit Ihrer Attribute zusammenfassen?
- Image Captioning - Wie lassen sich die klassifizierten Objekte in Beziehung setzen?

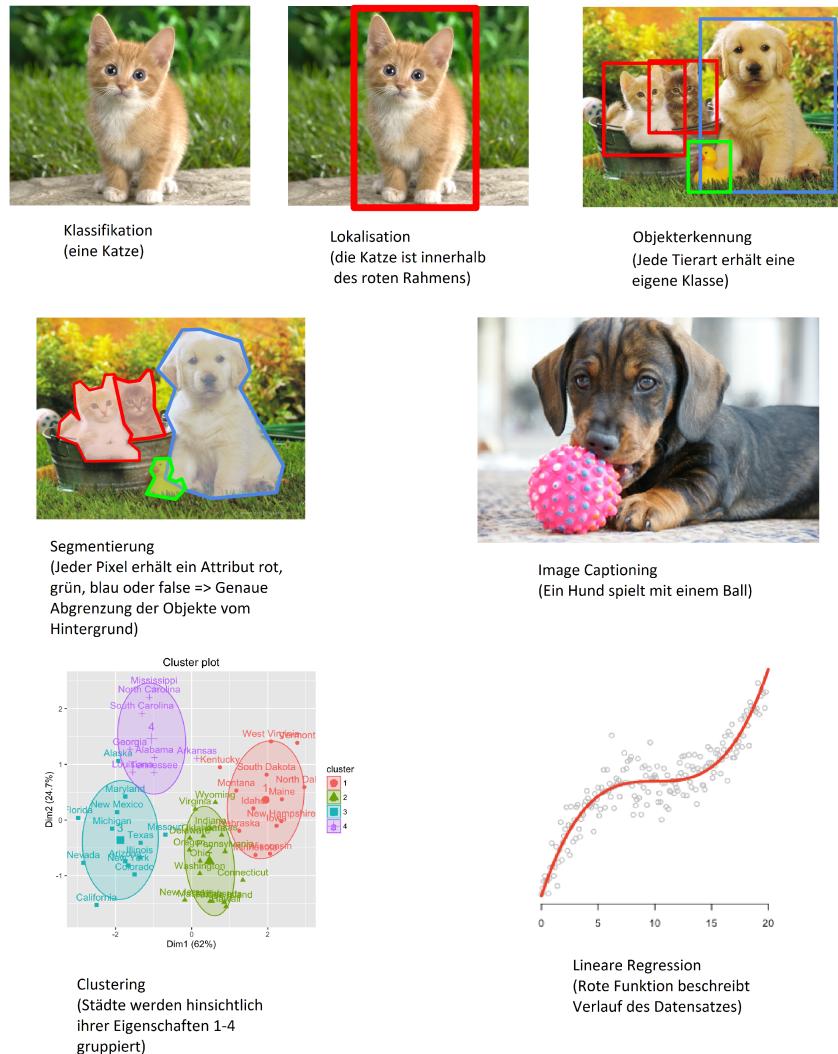


Abbildung 3.1: Vergleich verschiedener Anwendungszwecke von Deep Learning

### 3.2.1 Linear Classifier

Um bestimmen zu können, wie gut ein Bild  $x$  ( $x_i$  sind die einzelnen Pixelwerte) zu einer Klasse  $k_j$  passt, muss mithilfe einer Funktion  $f$  ein numerischer Vergleichswert bestimmt werden. Die Funktion  $f$  wird auch Score-Funktion genannt:

$$f(x_i; W, b) = W \cdot x_i + b$$

Der Parameter  $W$  ist die sogenannte Gewichtsmatrix. Sie besitzt die Dimensionalität  $i_{max} \times k_{max}$ . Der Parameter  $b$  heißt Bias und besitzt die Dimensionalität  $k_{max} \times 1$ . Er hat die gleiche Funktion wie die Gewichtsmatrix, ermöglicht aber eine zusätzliche additive Änderung beim Lernen. Im nächsten Unterkapitel wird die genauer Erläutert, wie dies zu interpretieren ist.

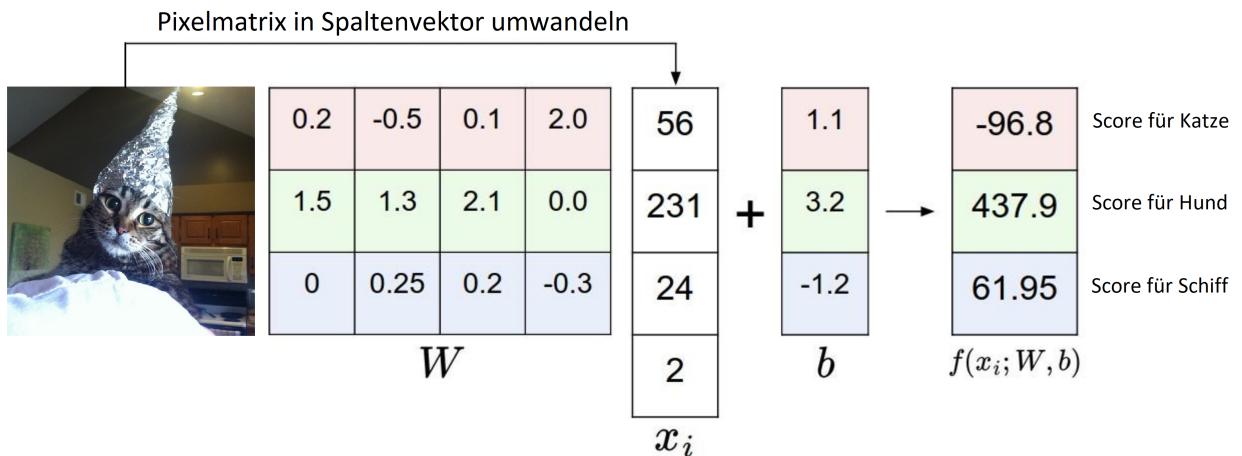


Abbildung 3.2: Interpretation der Score-Funktion anhand eines Beispiels

Wenn man ein Bild mit  $i_{max}$  Pixeln als  $i_{max}$  Dimensionalen Vektor auffasst, dann lässt sich ein Classifier als Hyperebenenseparator dieses Vektorraumes interpretieren. Je höher die Score Funktion für eine Klasse, desto geringer ist der Abstand zum Untervektorraum mit entsprechenden zur Klasse zugehörigen Mustern.

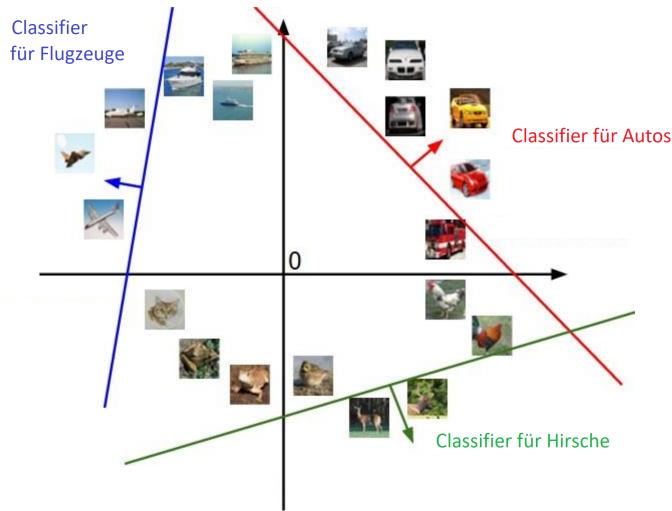


Abbildung 3.3: Hirsch-Auto-Flugzeug-Hyperraum mit Ebenenseparatoren

Äquivalent zur Score-Funktion definiert man auch eine sogenannte Loss-Funktion. Je besser ein Datum einer bestimmten Klasse zugeordnet werden kann, desto besser sind die Parameter der Gewichtsmatrix und des Bias und desto kleiner der Wert der Loss-Funktion. Im Rahmen eines Optimierungsprozesses soll die Loss-Funktion durch Änderung der Gewichte minimiert werden. Zwei häufig verwendete Loss-Funktionen für Daten sind:

Der gesamte Loss eines Optimierungsproblems besteht jedoch nicht nur aus dem oben genannten Datenloss, sondern des Weiteren aus dem Regularisierungs-Loss.

### 3.2.2 Regularisierer

Ein Regularisierer soll verhindern, dass sich die Gewichte immer nur hinsichtlich einer bestimmten Klasse verändern. Zum Beispiel kann es sein, dass ein Netzwerk, das ohne Regularisierer trainiert wurde zwar gut Katzen erkennen kann, jedoch keine Hunde. Das Ziel ist hierbei also, ein Netzwerk zu erzeugen, dass die Generalisierung "Tier" versteht und darüber hinaus zwischen verschiedenen Tierarten unterscheiden kann. Eine Möglichkeit, dies zu realisieren, bieten Bestrafungsterme:

$$R(W) = \sum_k \sum_l W_{k,l}^2$$

Dieser Bestrafungsterm geht ebenfalls in die Loss-Funktion mit ein. Die Quadrierung der Gewichte sorgt dafür, dass zuvor bereits deutlich größere Gewichte in der Loss-Funktion umso mehr berücksichtigt werden. Im Laufe des künftigen Trainingsprozesses können Ausreißer effektiv erkannt und vermieden werden. Die Loss-Funktion wird nur dann minimal sein, wenn sich die Gewichte nicht zu stark voneinander unterscheiden.

Die gesamte Loss-Funktion lautet also:

$$L_{ges} = \frac{1}{N} \sum_i L_i + \lambda R(W)$$

### 3.2.3 Gradientenabstieg

Um die Loss-Funktion zu minimieren, müssen die Gewichte  $W$  aktualisiert werden. Hierzu berechnet man den negativen Gradienten (Partielle Ableitungen der Loss-Funktion nach den Gewichten). Auf der Ebene, welche die Loss-Funktion aufspannt, zeigt der Gradient in Richtung des steilsten Abstiegs.

### 3.2.4 Multi-Layer Neural Network

Ein Multi-Layer Neuronales Netzwerk ist ein gerichteter azyklischer Graph, wobei die Knoten in Ebenen angeordnet werden. Die erste Knotenebene besteht aus den Eingabedaten. Im Bildbeispiel wäre jeder Knoten ein Pixelwert. Die Kanten des Graphen entsprechen den Werten der Gewichtsmatrix  $W$ . Die Knoten der Hidden-Layers unter der Output-Layer lassen sich als Neuronen interpretieren, welche durch Aktivierung oder Deaktivierung bestimmte Muster der Eingabedaten lernen. Ob ein Neuron aktiviert wird oder nicht, bestimmt die Aktivierungsfunktion  $g$ . Das Ergebnis wird an eine weitere Neuronenschicht weitergeleitet und die gelernten Muster somit weiter verfeinert.

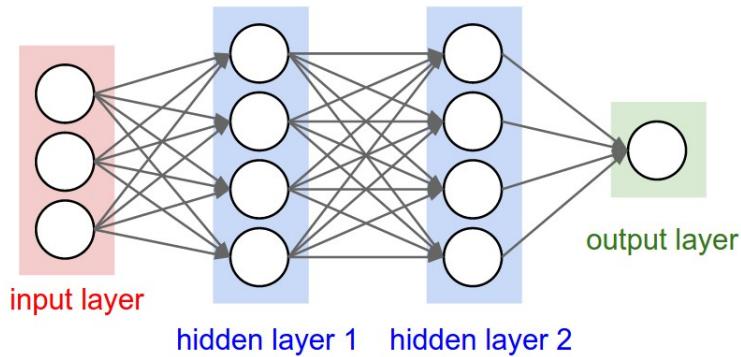


Abbildung 3.4: Multi-Layer NN. mit 2 Hidden-Layers, 32 Gewichten, 9 Biases und 9 Neuronen

Die Score-Funktion für dieses Netzwerk lautet:

$$y(x) = W_3^T \cdot g[W_2^T \cdot g[W_1^T \cdot x + b_1] + b_2] + b_3$$

### 3.2.5 Trainieren eines Netzwerks mittels Backpropagation

Sei  $f$  eine Score-Funktion für ein Single Layer Neural Network mit nur einer Aktivierung.

$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$

Diese Aktivierungsfunktion nennt sich auch Sigmoidfunktion. Sie transformiert reelle Zahlen auf ein Intervall  $[0, 1]$ .

Das Ergebnis von  $f(x, w)$  mit  $w = (2, -1, -3)$  und  $x = (-1, -2)$  ist 0.73. Die Ableitung von  $\frac{1}{x}$  ist  $-\frac{1}{x^2}$ . Anschließend rechnet man  $-\frac{1}{1.37^2} \cdot 1 = -0.53$ . Die Ableitung von  $x + 1$  ist 1. Also rechnet man  $1 \cdot -0.53 = -0.53$ . Die Ableitung von  $e^x$  ist  $e^x$ . Man rechnet  $e^{-1} \cdot 0.53 = -0.2$ . Diese Vorgehensweise entspricht der Anwendung der Kettenregel. Fährt man fort, erhält man

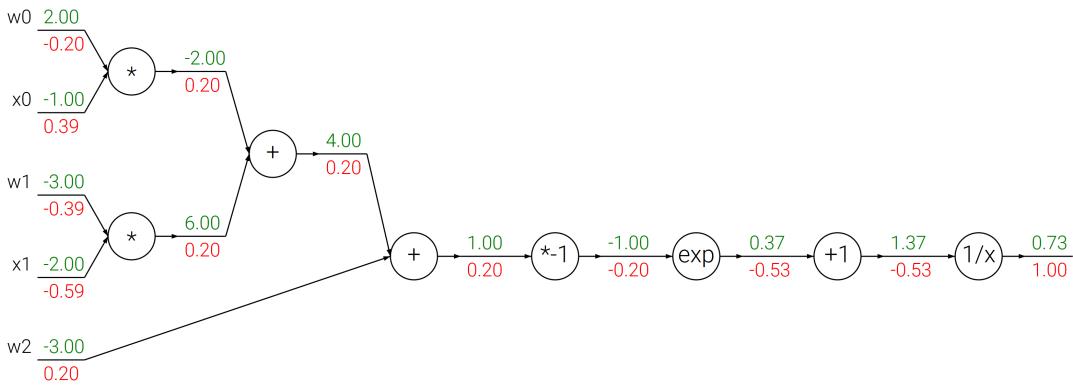


Abbildung 3.5: Beispiel für Backpropagation anhand der Sigmoid-Funktion

Gewichtsänderungen  $W_{exch} = (-0.2, -0.4, 0.2)$ . Diese Änderungen subtrahiert man von den alten Gewichten mit einem Parameter  $step$ :  $W := step \cdot W_{exch}$ . Dabei gibt Lernrate  $step$  an, wie stark die Änderung der Gewichte ausfallen soll. Eine zu große Änderung kann bedeuten, dass man über das Minimum der Loss-Funktion hinaus springt. Sind die Änderungen zu klein, so muss der Gradient oft berechnet werden, was sehr lange dauern kann. Das Ergebnis der Score-Funktion für  $W' = (1.8, -3.4, -2.8)$  ist 0.9. Das Netzwerk klassifiziert nun also besser.

Neben der Sigmoidfunktion gibt es noch viele andere Aktivierungsfunktionen. So hat die Maximumsfunktion (Rectifier Linear Unit - ReLU) gegenüber der Sigmoidfunktion den Vorteil, dass bei relativ hohen Werten nicht alle Neuronen aktiviert werden und es somit weiterhin zu einem Lerneffekt kommt. Ein weiterer Vorteil des Rectifiziert ist die schnellere Konvergenz des Gradienten. So stellt sich bereits nach wenigen Trainings ein starker Lerneffekt ein. Sind die gewichtete jedoch zu Beginn sehr klein, so ist die Gefahr groß, dass bei einer weiteren Verkleinerung alle Neuronen deaktiviert werden.

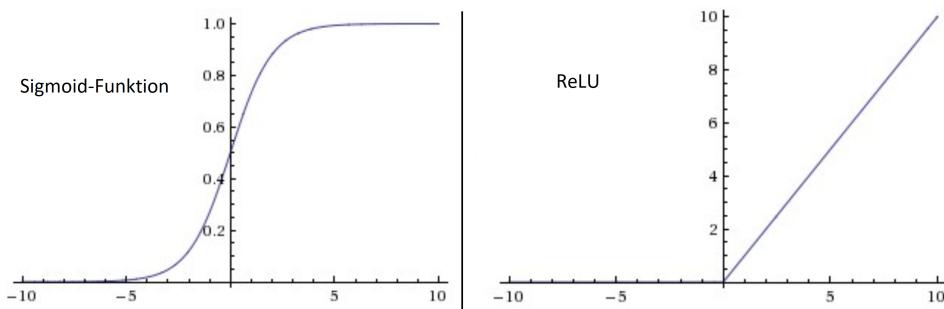


Abbildung 3.6: Vergleich zwischen Rectifier und Sigmoid

### 3.2.6 Parameterupdates<sup>2</sup>

**Batch Gradient Descent (BGD)** Normaler einmaliger Updateprozess wie oben beschrieben. Diese Variante ist sehr langsam, da für jedes Update der Gradient für den gesamten Datensatz berechnet werden muss.

$$W = W - \eta \nabla_W L(W)$$

**Stochastic Gradient Descent (SGD)** In diesem Fall werden die Gewichte nach jedem Trainingsbeispiel aktualisiert. Dies beschleunigt den Trainingsprozess und verhindert Redundanzen:

$$W = W - \eta \nabla_W L(W; x_i)$$

Zur Beschleunigung des Gradientenabstiegs wurden ebenfalls Verfahren entwickelt:

- **Momentum** Falls der Gradient zu groß ist, wird das Ziel beim aktualisieren der Gewichte übersprungen. Diese Methode beschleunigt den Gradientenabstieg, indem die Anzahl der übersprungenen Minima reduziert wird (quantitative Oszillationsreduktion). Hierzu wird der Update-Vektor des vorherigen Schrittes durch einen Parameter  $\gamma$  gewichtet und zum neuen Update-Vektor addiert:

$$v_t = \gamma v_{t-1} + \eta \nabla_W L(W)$$

$$W = W - v_t$$

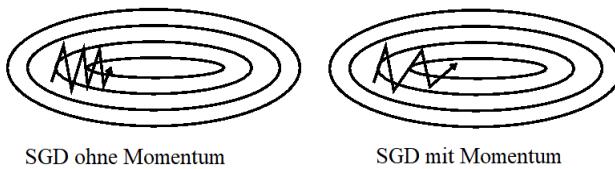


Abbildung 3.7: Momentum verringert die Oszillation der Loss-Funktion

- **Adagrad** Dieses Verfahren findet für jeden Parameter in jedem Updateschritt eine individuelle Lernrate. Dies bewirkt, dass das Minimum bei starker Steigung nicht übersprungen wird. Die Oszillation wird qualitativ reduziert.

$$W_{t+1,i} = W_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \nabla_W L(W_{t,i})$$

Wobei  $G_t$  eine Diagonalmatrix ist und das Element  $i, i$  ist die Summe der Quadrate der Gradienten bezüglich  $W_t$  bis Zeitschritt  $t$ . Das  $\epsilon$  verhindert eine Nulldivision.

<sup>2</sup><http://ruder.io/optimizing-gradient-descent/>

### 3.2.7 Convolutional Neural Network

Ein Convnet ist ein Neuronales Netzwerk, das speziell für das Klassifizierungproblem geeignet ist. Ein Standardnetzwerk besteht aus folgenden vier Layern:

- **1. Convolution Layer** Dies ist ein Filter  $W$ , der Dimension  $F \times F \times c$ , wobei das Bild  $P$  die Dimension  $N \times N \times c$  hat und  $N > F$ . Die neue Größe (Höhe/Breite  $N$ ) des Ausgabebildes nach der Convolution beträgt  $\frac{N-F}{stride+1}$ , wobei der Stride die Anzahl der Gitterzellen ist, die der Filter jedes mal weiter springt. Als Padding bezeichnet man die Anzahl der Gitterreihen, die um das Ursprungsbild herum virtuell angefügt werden, um zu verhindern, dass das Bild durch die Convolution kleiner wird. Zu Beginn wird der Filter an die linke obere Ecke angesetzt. Anschließend wird der erste Wert des neuen Bildes berechnet.

$$\sum_{i=1}^F \sum_{k=1}^F P_{i,k} \cdot w_{i,k}$$

Danach wird der Filter um *stride* Gitterzellen nach rechts verschoben und es wird erneut aufsummiert.

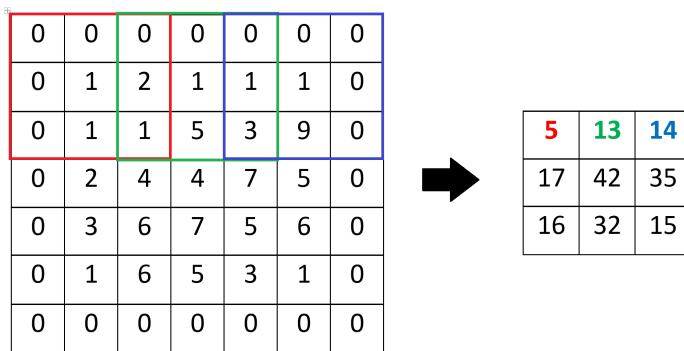


Abbildung 3.8: Convolution eines  $7 \times 7$  Bildes mittels  $3 \times 3$  Eins-Filter ( $w_{i,k} = 1$ ), Padding = 1 und Stride = 2

Jeder Filter erstreckt sich über die gesamte Tiefe des Bildes. Wird also ein  $B \times B \times 3$  Filter auf ein  $A \times A \times 3$  Bild angewandt, so hat das Ausgabebild die Dimension  $C \times C \times 1$  mit  $C = \frac{A-B}{stride+1}$ . Werden 10 Filter der Größe  $B \times B \times 3$  angewandt, so hat die Ausgabe die Dimension  $C \times C \times 10$ . Je mehr Filter angewandt werden, desto besser können Muster gelernt werden. Zu große und zu kleine Filter führen zu Over- und zu Underfitting.

- **2. ReLU Layer** Diese Layer übernimmt das Aktivieren der einzelnen Gewichte mit z.B.  $\max(0,x)$ .
- **3. Pooling Layer** Diese Layer reduziert die Ausgabe nach der Filterung und Aktivierung auf wesentliche Merkmale des Bildes und verkleinert die Anzahl der Pixel, die für die Speicherung dieser Merkmale notwendig sind. Zum Einsatz kommt meistens das Max-Pooling. Hierbei wird wie bei der Convolution eine Maske auf das Bild gesetzt. Das Maximum aller Bildwerte, die sich hinter der Maske befinden, ist der Wert des neuen Bildes nach dem Pooling.

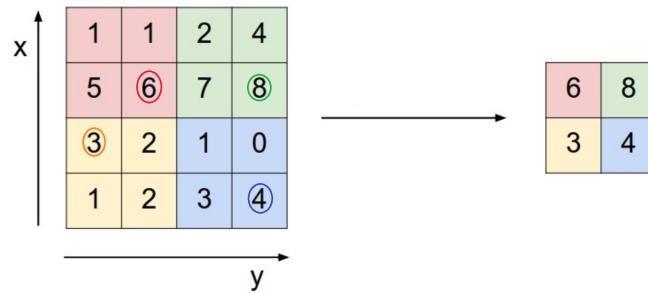


Abbildung 3.9: Pooling mit stride = 2

- **4. Fully Connected Layer** Diese Neuronen-Schicht soll die Ausgabe nach den oben genannten Schritten interpretierbar machen und auf Klassen-Scores abbilden. Jedes Neuron ist mit jedem aktivierten Gewicht davor liegenden Schicht verbunden.

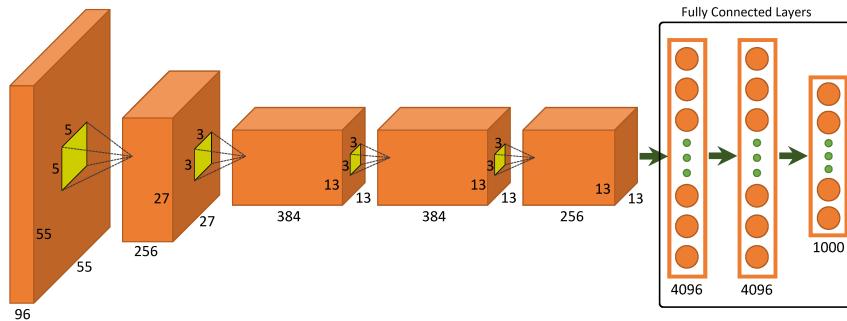
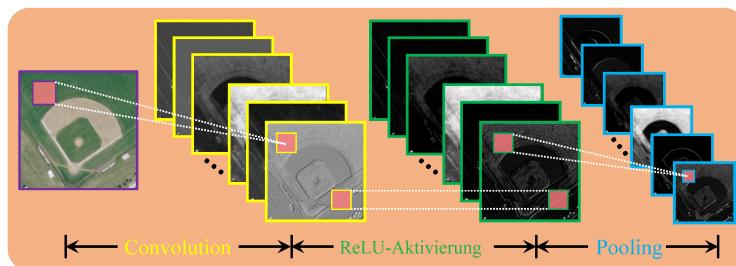


Abbildung 3.10: Gewöhnliches Convnet mit Fully-Connected-Layers zum Schluss

Abbildung 3.11: Verbildlichung der Schritte 1-3 durch Anwendung von Convolution auf ein Luftbild<sup>4</sup>

### 3.2.8 Recurrent Neural Network

Ein Recurrent Neural Network (RNN) ist ein zeitabhängiges neuronales Netz, mit welchem sich Zustände vorhersagen lassen. So lassen sich mit einem RNN z.B. folgende Probleme Lösen:

<sup>4</sup><http://www.mdpi.com/2072-4292/7/11/14680/htm>

- Image Captioning
- Sentimentanalyse (Stimmungserkennung im Text Mining)
- Sprachübersetzung, Textproduktion und Vorhersage von Worten
- Videoklassifikation
- Vorhersage von geologischen Aktivitäten

Ein Zustand  $h_t$  und eine Ausgabe  $y_t$  eines RNN zu einem bestimmten Zeitpunkt  $t$  berechnet sich wie folgt:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b1)$$

$$y_t = g(W_{hy}h_t + b2)$$

Hierbei enthalten die Gewichtsmatrizen  $W_{xh}$ ,  $W_{hh}$ ,  $W_{hy}$  die zu lernenden Parameter zwischen Input Layer, Hidden Layer und Output Layer. Die Funktionen  $f$  und  $g$  sind Nichtlinearitäten.

#### Beispiel:

Ein RNN wird mit einer gegebenen Sequenz ('h' 'e' 'l' 'l' 'o') trainiert, so dass es bei der Eingabesequenz  $x = ('h' 'e' 'l' 'l')$  die Sequenz ('e' 'l' 'l' 'o') und somit den Buchstaben 'o' korrekt vorhersagt.

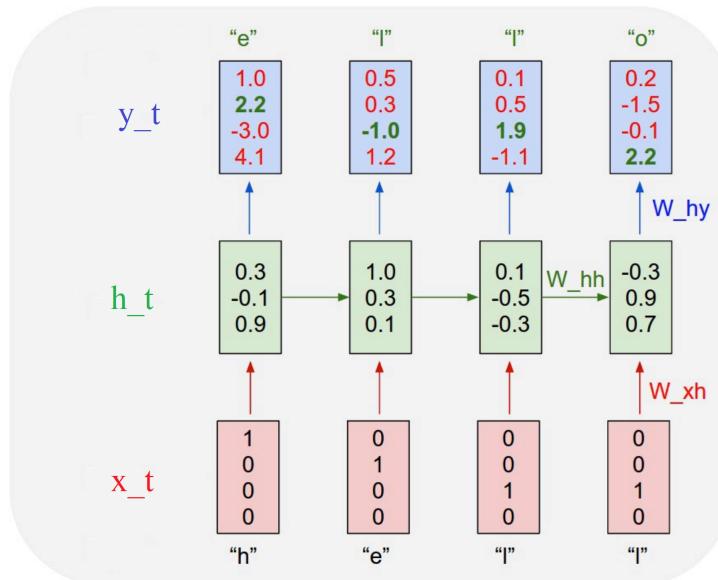


Abbildung 3.12: RNN vervollständigt das Wort *hello*

Die Gewichtsmatrix  $W_{xh}$  speichert, wie Buchstaben zu Hidden States umgewandelt und inkrementiert werden.  $W_{hy}$  lernt, welcher Ausgabebuchstabe zu gegebenem Hidden State passt und die Matrix  $W_{hh}$  akkumuliert und migriert alle bisherigen Eingaben im zeitlichen Verlauf zu einem gemeinsamen Hidden State.

### 3.2.9 Long Short Term Memory Netzwerke

#### Das Problem der Langzeitabhängigkeit

Angenommen, es soll ein RNN trainiert werden, welches das letzte Wort des folgenden Satzes vorhersagt: *Ich spreche fließend*. Offensichtlich ist das gesuchte Wort eine Sprache. Leider geht diese Information nicht aus dem Satz hervor, da der Kontext fehlt. Ein vorangestellter Satz *Ich bin ich Deutschland aufgewachsen*. grenzt die möglichen Worte stark ein. Ein RNN, wie oben beschrieben, kann jedoch nur Kurzzeitinformationen speichern. [Hoc91] Aus diesem Grund wurden spezielle Variante eines RNNs erfunden.

**LSTM-Zellen**

[Ola15]

### 3.3 Machine Learning in Geowissenschaften

#### 3.3.1 Herausforderungen und Chancen von Machine Learning (Kar+17)

##### Einleitung

Die moderne Geowissenschaft steht vor vielen Herausforderungen, wie den Prognosen zu Klimawandel, Luftverschmutzung, Naturkatastrophen, Ressourcenverbrauch oder den Risiken für Erdbeben, Landrutsch und Vulkanausbrüchen. Die Forschung an solchen Problemen macht die interdisziplinäre Arbeit sämtlicher Wissenschaften unumgänglich. Geowissenschaften haben sich in den letzten Jahrzehnten zu einer Big-Data Disziplin entwickelt. Angestoßen wurde diese Entwicklung durch sämtliche technologische Verbesserungen, wie dem Wachstum der Computerleistung. Aufwändige Simulationen und die Demokratisierung von Datenbeständen, deren öffentliche Verbreitung im Internet und vielseitige Beteiligungsmöglichkeiten trugen ebenfalls dazu bei. Die wachsende Verbreitung großer geowissenschaftlicher Datenbestände ermöglicht ein enormes Potenzial für Maschinelles Lernen.

##### Sammeln von Geodaten

Die Erde ist ein komplexes dynamisches System, bestehend aus Litosphäre, Biosphäre, Atmosphäre, Hydrosphäre. Einzelne Bestandteile dieses Systems (z.B. Ozeanschichten oder die Bodenbedeckung) befinden sich in einem ständigen Wandel und interagieren miteinander. Um Daten solcher Phänomene zu erheben, gibt es vor allem zwei Möglichkeiten:

- Messen mithilfe von Sensoren in/auf Satelliten, Flugzeugen, Ballons, Drohnen, Wetterstationen, Schiffen, Bojen.  
Sensorbasierte geowissenschaftliche Beobachtungen sind nicht uniform gerastert und beziehen sich häufig auf irreguläre Zeitintervalle (zum Beispiel schwankt die Position einer Boje mit der Zeit). Sie eignen sich jedoch Hervorragend zum sammeln von Daten wie Oberflächentemperatur, Luftfeuchtigkeit, Reflektionintensität, Chemischer Zusammensetzung der Atmosphäre, Strömungen und Drücke, Emissionen, seismischer Aktivität und der Oberflächengestalt der Erde. Diese Vielfalt an geologischen Eigenschaften erfordert bezüglich eines gegebenen Problems eine individuelle Zusammenstellung von relevanten Datensätzen. Die Datentypen müssen gegebenenfalls konvertiert und die Datenbestände interpoliert werden, um sie besser interpretierbar zu machen.
- **Ableiten aus mathematischen Modellen und Simulationen** Geologische Prozesse, ihre Interaktionen und Änderungen sind auf physikalische Gesetze zurückzuführen. Zum Beispiel sind Bewegungen von Wasser in der Liosphäre auf die Strömungsdynamic zurückzuführen. Ein Nachteil physikalischer Berechnungen auf Grundlage von Modellen für komplexe Systeme ist leider deren Ungenauigkeit. Dennoch eignen sie sich zur näherungsweisen Darstellung des zeitlichen Verlaufes von geophysikalischen Phänomenen. Ein weiterer Vorteil ist, dass Simulationen besonders große Datensätze erzeugen können, wenn Daten für große Zeitintervalle von Interesse sind. Diese ermöglichen dann wiederum eine Datenbasierte Analyse mittels Machine Learning.

**Herausforderungen** Leider ist die Nützlichkeit von Machine Learning für Knowledge Discovery häufig begrenzt. Geophysikalische Objekte sind häufig nicht klar definiert (keine klar definierten Grenzen) und ändern sich häufig. Daten zu solchen Objekten können ebenfalls unterschiedliche Auflösungen haben, rauschen, unvollständig oder ungenau sein. Auch kann die zeitliche Auflösung aus historischen Gründen stark variieren (z.B. ein Weltkrieg, in dem historische Datensätze zerstört

wurden). Auch liegen häufig nicht ausreichend Geländedaten vor. Die Herausforderungen lassen sich in 3 Hauptkategorien unterteilen:

- **Eigenschaften Geologischer Prozesse**

- **Amorphe Grenzen (Wellen, Flüsse, Stürme)** Segmentierungs- und Clusteringverfahren sowie Maßnahmen für Feature-Charakterisierungen sind notwendig.
- **Raumzeitliche Struktur** Für viele Machine Learning Methoden nimmt man an, dass beobachtete geophysikalische Eigenschaften nicht korrelieren und die erhobenen Daten gleichverteilt sind. Die Realität sieht anders aus. Benachbarte Orte sind stark korreliert (Wahrscheinlichkeit für Grasland ist in der Nähe eines Waldes größer als in einer Wüste). Änderungen (Wald => Wüste) bewirken Zustände, die für eine unbestimmbare Zeit persistieren (Die Wüste wird nicht regelmäßig zum Wald und umgekehrt), was auf Klimaveränderungen zurückzuführen ist. Zwei weit entfernte Orte können ebenfalls stark korrelierte Eigenschaften besitzen (z.B. Temperatur, Druck). Man nennt diese meist meteorologischen Korrelationen Telekonnektionen .
- **Hochdimensionalität** Die Erde ist eines der komplexesten bekannten Systeme mit einer extrem großen Anzahl an Variablen, die alle miteinander in sowohl räumlich als auch zeitlich relativ zur Größe der Erde winzigen Skalen miteinander wechselwirken. Für eine Verarbeitung und Speicherung von Daten solcher Systeme ist die Rechenleistung heutiger Computer nicht ausreichend.
- **Raumzeitliche Variabilität** Geologische Prozesse können stark schwanken, sowohl in kurzen Zeitintervallen (Jahreszeiten, Tidenhub), in langen Zeitintervallen (Polsprung, Präzession, Klimawandel) als auch räumlich (Gebirgsformationen, Vegetationszonen, Klimazonen). Es ist sehr schwierig ein Modell zu trainieren, dass alle diese Prozesse vereint. Eine lokale und zeitliche Begrenzung der Datensätze ist zwingend erforderlich.
- Seltene Phänomene - Seltene Ereignisse wie z.B. Vulkanausbrüche, Tsunamis und Erdbeben verfälschen das Modell, da sie nur für ein Training auf viel größeren Zeitskalen geeignet sind. Aus diesem Grund müssen sie erkannt und aus dem Modell herausgerechnet werden. Dies ist nahezu unmöglich, da es hierzu keine ausreichenden Erfahrungswerte gibt.

- **Sammeln von Geodaten**

- Daten mit verschiedenen Auflösungen - Beispiel: Zur Beurteilung von Waldbränden müssen Bilder aus Luftaufnahmen und Satellitenbilder miteinander kombiniert werden. Die Flugzeugbilder haben eine höhere räumliche Auflösung, die Satellitenbilder wurden jedoch in regelmäßigen Zeitabständen aufgenommen. Aus diesem Grund müssen Interpolations- oder Upsamplingmethoden entworfen werden, um die beiden Arten von Bildern vergleichbar zu machen.
- Rauschen, Unvollständigkeit, Ungenauigkeit - Viele Geodatensätze sind unvollständig oder rauschen, weil z.B. Sensoren temporär ausgefallen sind oder unter verschiedenen Wetterbedingungen Messungen durchgeführt haben. Manche Daten sind erst dann interpretierbar, wenn Sie mit einem mathematischen Modell kombiniert werden. Dieses kann die Interpretierbarkeit jedoch ebenfalls beeinflussen.

- **Mangelhafte Datensätze**

- **Kleine Sample-Größe** Viele Datenbestände sind niedrigfrequent und extrem ungenau, wenn sie aus einer Zeit stammen, in welcher es entweder keine oder nur wenige Messinstrumente gab. Auch gibt es Orte, an denen es nur schwer möglich ist (zeitlich hochfrequente) Messungen durchzuführen (Bohrkernanalyse in Antarktis o.ä., Bäume

mit ausreichendem Alter für Jahresringe). Eisbohrkerne aus der Antarktis lassen zudem keine Rückschlüsse über die Klimatischen Verhältnisse in anderen Regionen zu.

- **Mangelhafte gelabelte Geländedaten** Oft liegen Geländedaten in mangelnder Qualität vor. Dies liegt daran, dass sie nur mit Zeit- und Kostenintensiven Maßnahmen zu beschaffen sind. Eine geringe Datenqualität führt zu einem langsamem Trainingsprozesses und zu Unter- oder Überanpassung des Modells, was die Aussagekraft stark einschränkt.

Aus beiden oben genannten Gründen müssen Trainingsmodelle entwickelt werden, die mit kleineren Datensätzen zurecht kommen.

Maschinelles Lernalgorithmen können dazu beitragen, Geowissenschaftliche Objekte und Ereignisse zu Charakterisieren und somit helfen, das Erdsystem besser zu verstehen. Während traditionelle Ansätze auf handgefertigten Algorithmen basieren, können Machine Learning Algorithmen mit ihrer automatischen Mustererkennung die Berechnungszeit deutlich verkürzen. Eine große Herausforderung stellt die Charakterisierung von physikalisch ungenau definierten Objekten dar. Unsupervised Learning kann dabei helfen, anomale Objekte aufzuspüren (z.B. Landminen).

Ein weiterer großer Vorteil von Machine Learning ist die Erzeugung von Geodaten aus nur schwer beobachtbaren Prozessen (z.B. Methanausstoß und Konzentration in der Atmosphäre). Supervised Learning kann verwendet werden, um Fernerkundungsdaten zu analysieren und daraus Aussagen über das Ökosystem abzuleiten (z.B. Gesundheit der Vegetation oder Wasserqualität). Eine besondere Herausforderung ist dabei die Heterogenität solcher Daten, wie bereits weiter oben beschrieben. Eine mögliche Lösung sind Multi-Task-Learning-Frameworks, welche Datensätze zuerst in homogene Partitionen unterteilen (hierarchisches Clustern), und dann auf jeder dieser Partitionen einzeln trainieren. Dieses Vorgehen ist eine Regularisierungstechnik, welche die Überanpassung des Modells verhindern soll. Folgende Abbildung zeigt die Verbesserung der Genauigkeit der geschätzten Waldbedeckung vier brasilianischer Staaten, wobei die rot markierten Bereiche die Residuen darstellen.

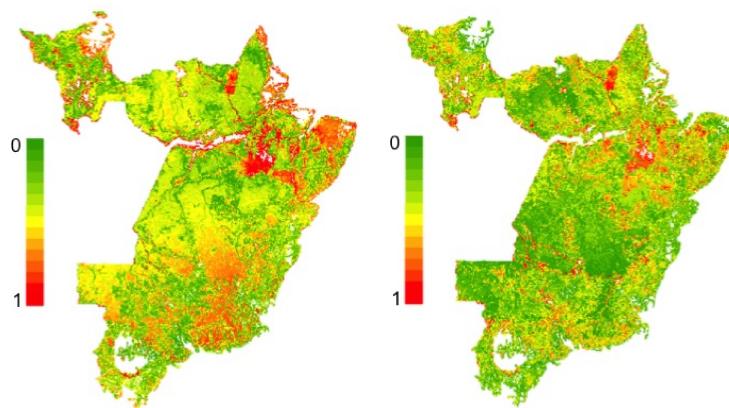


Abbildung 3.13: Schätzung der Waldbedeckung in Brasilien<sup>5</sup>

---

<sup>5</sup>A. Karpatne, Z. Jiang, R. R. Vatsavai, S. Shekhar, and V. Kumar, Monitoring Land-Cover Changes. IEEE Geoscience and Remote Sensing Magazine, 2016

### Nichtstationarität, Heterogenität und Mangeldaten

Zum Umgang mit der Nichtstationarität (= Variable folgt keinem konstanten Wert) von Klimadaten wurden lernende Algorithmen entworfen, welche die Vorhersagen verschiedener Klimamodelle kombinieren. Statt eines Mittelwertes über die Klimamodelle berücksichtigt der selbstlernende Algorithmus zusätzlich die nichtstationären Dateneigenschaften und erzeugt wesentlich genauerer Vorhersagen.

Heterogene und minderqualitative Daten können mithilfe von *Adaptive Ensemble Learning* und *Label Refinement* besser analysiert werden.

Zur Näherung von Geophysikalischen Variablen mit Daten geringer zeitlicher Auflösung und fehlender Labels können folgende Techniken verwendet werden: - Regularisierer (für zeitl. Auflösung) - Semi-Supervised-Learning (fehlende Labels) - Active Learning (genauere Ergebnisse für Berechnungsprobleme) - Unsupervised Learning (bessere Näherung für geophysikalische Größen und Kartierung von Flächenänderungen durch z.B. Insektensterben, Waldabholzung und Ackerlandumwandlung)

### Techniken für Langzeitvorhersagen

Eine Möglichkeit für Langzeitvorhersagen war bisher die Verwendung von physikalischen Modellsimulationen. Diese können jedoch auch als Zeitreihen-Regressions-Problem aufgefasst werden. Mögliche Methoden zur Lösung solcher Probleme sind z.B.:

- Exponential Smoothing Techniques
- Autoregressive Integrated Moving Average (ARIMA) Modelle
- State-space Modelle
- Hidden Markov Modelle
- Kalman Filter

Transfer Learning - Ein zu trainierendes Modell für ein Problem mit wenig Daten soll mithilfe eines zuvor trainierten Modells mit vielen Daten das Problem besser lösen.

### Relationen und Kausalität

Geophysikalische Zusammenhänge (z.B. Telekonnektionen und Dipole), sollen sich mithilfe datenbasierter Ansätze besser verstehen lassen. Man erhofft sich durch sie die Entdeckung neuer Korrelationsmuster. Darüber hinaus können graphenbasierte Repräsentationen von Klimadaten mithilfe von Clustering und Mustererkennung besser analysiert werden. Eine besondere

Herausforderung bei der Korellationserkennung ist der besonders große Suchraum mit all seinen raumzeitlichen Objekten und dynamischen, rauschenden und unvollständigen Geodaten. Es herrscht ein großer Bedarf an neuen Ansätzen, die gleichzeitig sowohl Zusammenhänge als auch die dazugehörigen interagierenden Objekte erkennen.

Ursache-Wirkungszusammenhänge zu entdecken ist eine weitere wichtige Aufgabe in den Geowissenschaften. Ein häufig eingesetztes Tool für die Analyse von solchen kausalen Zusammenhängen ist die multivariate Grangeranalyse mittels Vektorautoregression. Auf diese Weise können z.B. Sturmverläufe vorhergesagt werden. Weitere kaum erforschte Möglichkeiten sind das Reinforcement Learning und stoastische Anstze der dynamischen Programmierung zur Lösung von Entscheidungsproblemen.

### Deep Learning

Neuronale Netze haben die Eigenschaft, komplexe Features mithilfe der Verknüpfung weniger komplexer Features darzustellen. In Kombination mit dem Training großer Datensätze und der

Fähigkeit, Fehler an den Nodes der Hidden-Layers zu minimieren, haben neuronale Netze weite Felder im Bereich Machine Learning revolutioniert. Darunter auch Supervised, Semi-Supervised, und Reinforcement Learning. Häufig werden neuronale Netze eingesetzt, wenn es schwierig ist, mittels handgeschriebener Algorithmen die wirklich relevanten Features aus einem komplexen Datensatz (wie es auch Geodatensätze sind) zu extrahieren. Geophysikalische Fragestellungen und Probleme haben viele Ähnlichkeiten zu den Themen, die im Bereich Computer Vision und Spracherkennung behandelt werden.

Während man ein Convolutional Neural Network einsetzt, um auf einem Bild eine Katze zu klassifizieren, kann man dieses auch verwenden, um Wetterphänomene wie Tornados auf Satellitenbildern zu erkennen.

Rekurrente Neuronale Netze mit Longshort-Term-Memory Zellen (LSTM) können z.B. genutzt werden, um zeitlich dynamische Plantagen mittels Fernerkundungsdaten kartografieren. RNNs besitzen die Eigenschaft besitzen, zeitlich vergangene Information zu speichern und in zukünftige Vorhersagen mit einzubeziehen. Aus diesem Grund eignen Sie sich zur Vorhersage von geologischen Ereignissen mit angemessener Vorlaufzeit.

Deep Learning kann bisher nur bei ausreichend gelabelten Daten zum Einsatz kommen. Aus diesem Grund herrscht hier ein Bedarf an neuen Verfahren, die mit nur wenigen Daten zureckkommen.

### **Fazit**

Die Forschung der letzten Jahre hat gezeigt, dass weder ein reiner Datenansatz noch ein reiner mathematischer Modellansatz ausreichend ist, um Knowledge Discovery effizient betreiben zu können. Aus diesem Grund sollten zukünftige physikalische Erkenntnisse möglichst früh und tief in Datenwissenschaftlichen Ansätze einbezogen werden. Auf diese Weise lässt sich auch die Wahrscheinlichkeit für Overfitting reduzieren, vor allem bei mangelnden Trainingsdaten.

#### **3.3.2 Einsatz in der Praxis - Beispiele**

##### **3.4 Tensorflow**

##### **3.5 Erstes eigenes CNN**



## 4. Clusteringverfahren

### 4.1 Ziel

### 4.2 Eindeutiges Clustering

#### 4.2.1 Partitionierende Verfahren und ihre Nachteile

#### 4.2.2 Dichtebaserte Verfahren

#### 4.2.3 Hierarchische Verfahren

#### 4.2.4 Outlier Detection

### 4.3 Fuzzy-Clustering

#### 4.3.1 Probabilistisches und Possibilistisches Clustering

FCM und PFCM

Voraussetzungen für die Anwendung auf Geodaten

Eigener Algorithmus (noch ohne Name)

### 4.4 CVI

#### 4.4.1 NPC

#### 4.4.2 FHV

#### 4.4.3 Otsu-Binarisierung

#### 4.4.4 VAT-Algorithmus

### 4.5 Clustering auf unvollständigen Daten

### 4.6 Einsatz in der Praxis und Software-Tools





## 5. Fazit und Ausblick

### 5.1 Ausblick - Mein Thema für die Masterarbeit





## Literaturverzeichnis

- [FPS96] Usama Fayyad, Gregory Piatetsky-Shapiro und Padhraic Smyth. *Knowledge Discovery and Data Mining: Towards a Unifying Framework*. 1996. URL: <https://www.aaai.org/Papers/KDD/1996/KDD96-014.pdf> (siehe Seite 9).
- [Hoc91] Josef Hochreite. *Untersuchungen zu dynamischen neuronalen Netzen*. 1991. URL: <http://people.idsia.ch/~juergen/SeppHochreiter1991ThesisAdvisorSchmidhuber.pdf> (siehe Seite 27).
- [Kar+17] Anuj Karpatne u. a. *Machine Learning for the Geosciences: Challenges and Opportunities*. Version 1. 13. Nov. 2017. arXiv: 1711.04708 (siehe Seite 28).
- [Lar+15] David J. Lary u. a. “Machine learning in geosciences and remote sensing”. In: *Geoscience Frontiers* (2015), Seiten 3–10 (siehe Seite 17).
- [Mit97] Tom M. Mitchell. *Machine Learning*. McGraw-Hill Education, 1997 (siehe Seite 17).
- [Ola15] Christopher Olah. *Understanding LSTM Networks*. 2015. URL: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (siehe Seite 27).