



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

## Proseminar Softwareentwicklung in der Wissenschaft 2021

# Deep Learning in der Wissenschaft

vorgelegt von

Jona Zantz

Fakultät für Mathematik, Informatik und Naturwissenschaften  
Fachbereich Informatik  
Arbeitsbereich Wissenschaftliches Rechnen

Studiengang: Informatik

Matrikelnummer: 7401144

Betreuer: Jakob Lüttgau

Tobias Finn

Hamburg, 2021-08-30

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>3</b>
<b>2</b>	<b>Methoden - Künstliche Intelligenz</b>	<b>4</b>
2.1	Deep Learning und Neuronale Netzwerke . . . . .	4
2.2	Lernprozess . . . . .	5
<b>3</b>	<b>Methoden - Erdbeben</b>	<b>7</b>
3.1	Entstehung . . . . .	7
3.2	Nachbeben . . . . .	7
<b>4</b>	<b>Vorhersage von Nachbeben</b>	<b>8</b>
4.1	Klassische Methoden . . . . .	8
4.2	Deep Learning Ansatz . . . . .	9
<b>5</b>	<b>Zusammenfassung</b>	<b>11</b>
	<b>Literaturverzeichnis</b>	<b>12</b>

# 1 Einführung

Diese Ausarbeitung beschäftigt sich mit dem Thema der Anwendung von Deep Learning für wissenschaftliche Arbeiten am Beispiel von Erdbeben. Hierfür baue ich auf das Paper *Deep Learning of aftershock patterns following large earthquakes* von DeVries et. al. [3] und einem auf dieses bezogenen Brief *One neuron versus deep learning in aftershock prediction* von Broccardo und Mignan [7] auf. Diese beschäftigen sich mit dem Vergleich von Deep Learning und anderen herkömmlichen Methoden wie dem Coulomb-Failure-Stress-Change-Kriterium zur Vorhersage von Nachbeben.

Für meine Ausarbeitung stelle ich das Konzept des Deep Learnings vor, sowie die Entstehung von Erdbeben und den daraus resultierenden Nachbeben. Auf dieser Basis kann die Genauigkeit von verschiedenen Vorgehensweisen und ihre Anwendungsmöglichkeiten in der Wissenschaft verglichen werden.

Anschließend versuche ich aus den Erkenntnissen zu Nachbeben auf allgemeine Eigenschaften von Deep Learning zu schließen und die daraus folgenden Vor- und Nachteile für eine wissenschaftliche Nutzung.

Die genannten Studien verwenden Daten aus der Earthquake Source Model Database, welche aus stress-change Tensoren bestehen. Diese Tensoren bestehen aus neun Komponenten, welche zusammen die Belastung an einem bestimmten Punkt abbilden.

# 2 Methoden - Künstliche Intelligenz

## 2.1 Deep Learning und Neuronale Netzwerke

Künstliche Intelligenz beschreibt den Versuch menschliches Denken in Form von Programmen umzusetzen. Auch wenn dies nicht der allgemeinen Vorstellung entspricht, bedeutet dies, das technisch gesehen jedes Programm, das eine Fallunterscheidung vornimmt, *intelligent* ist.

Das was Allgemein als KI verstanden wird fällt zumeist in die Unterkategorie des maschinellen Lernens. Bei diesem wird versucht einem Computer beziehungsweise einem Programm ein bestimmtes Vorgehen beizubringen ohne direkte Vorgaben in Form von programmierten Anweisungen zu geben. Maschinelles Lernen kann grob in zwei Klassen eingeteilt werden, überwachtes und unüberwachtes Lernen. Bei überwachtem Lernen wird als Grundlage ein Datensatz verwendet, in dem in der Vorbereitung allen Eingabewerten ein Ausgabewert zugeordnet wurde. Dieser Datensatz wird in Trainingsdaten und Testdaten aufgeteilt, die Trainingsdaten dienen zum tatsächlichen Training, die Testdaten zur Überprüfung des Lernprozesses. Auch bei unüberwachtem Lernen ist die Voraussetzung ein Datensatz, allerdings ist bei diesem nicht bekannt, wie die Ausgabewerte für die Eingabe aussehen sollen. Hierfür gibt es dann also Algorithmen, welche optimale Gruppierungen der Daten herausarbeiten können.

Das Deep Learning oder mehrschichtiges Lernen setzt beide dieser Verfahren mit künstlichen Neuronalen Netzen um. Das *Deep* bezieht sich hierbei auf die Tiefe eines solchen Netzes, welches aus verschiedenen Ebenen (Layern) aufgebaut ist.

Ein Neuronales Netz ist ein Konstrukt, welches versucht das menschliche Gehirn nachzustellen. Hierbei werden einzelne Neuronen als Knoten simuliert, welche eingehende und ausgehende Verbindungen besitzen. Diese sind in Layern strukturiert, wobei ein Neuron mit allen Neuronen des vorigen Layers und denen des unmittelbar folgenden Layers verbunden sind. Der erste Layer besitzt keine Eingaben sondern ist selbst der sogenannte Input-Layer, diese Neuronen erhalten ihre Werte aus den Daten. Der letzte Layer heißt Output-Layer und gibt die Ergebnisse des Netzwerks aus. Zwischen Input- und Output-Layer liegen die Hidden-Layer, in diesen finden die eigentlichen Berechnungen statt. Der Aufbau eines solchen Netzwerks kann in Abbildung 2.1 gesehen werden. (vgl. [2][8])

Um von einer Eingabe zu einer Ausgabe zu gelangen müssen die Neuronen in den verschiedenen Layern aktiviert werden. Bei der Aktivierung werden zuerst die eingehenden Werte mit der für jede Verbindung spezifischen Gewichtung aufsummiert und mit einem Bias verrechnet (vgl. [6])

$$y = x_1W_1 + x_2W_2 + \dots + x_nW_n + B.$$

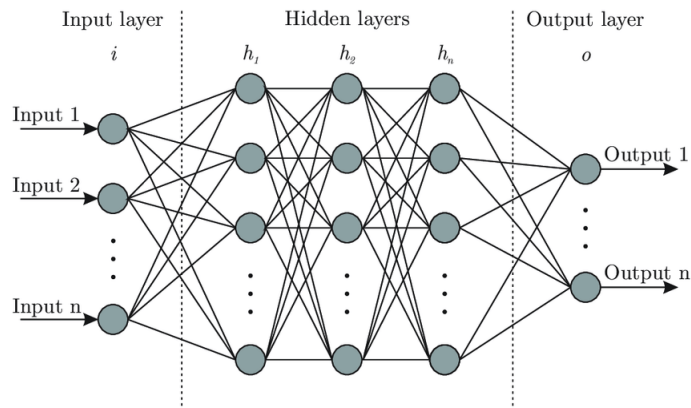


Abbildung 2.1: Schematische Darstellung eines Neuronalen Netzwerks [1]

Die Aktivierung erfolgt dann aufgrund dieses Wertes. Es gibt verschiedene Ansätze für die Aktivierungsfunktion. Ein sehr intuitiver Ansatz wäre, sobald die Eingabewerte größer als 0 sind, wird ein Signal ausgegeben, ansonsten nicht (siehe Abb. 2.2a). Eine andere Vorgehensweise wäre den Aktivierungswert in einen Bereich zwischen zum Beispiel 0 und 1 oder -1 und 1 zu normieren, dies geschieht mit Sigmoid-Funktionen (Abb. 2.2b) oder einem Tangens-Hyperbolicus (Abb. 2.2c). (vgl. [8]) Dies hat den Vorteil, dass die Aktivierungsfunktion stetig und differenzierbar ist, was für den Lernprozess von Vorteil ist.

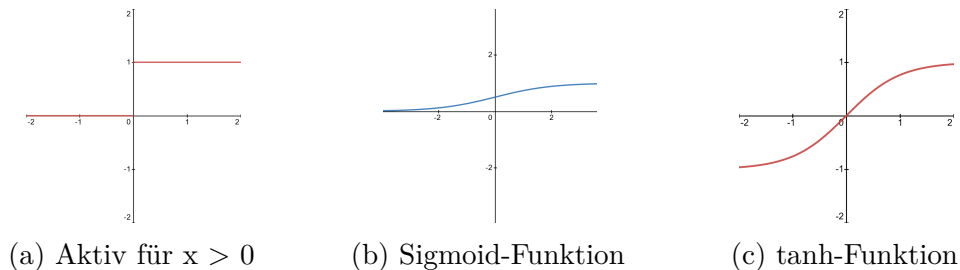


Abbildung 2.2: Beispiele für Aktivierungsfunktionen

## 2.2 Lernprozess

Der Lernprozess eines Neuronalen Netzwerks baut auf den Trainingsdaten auf. Wie genau oder „gut“ ein Netzwerk ist, wird mit einer Loss-Funktion berechnet [8]. Diese nimmt die Differenzen zwischen erwarteten Ausgabe-Werten und den tatsächlichen Werten und summiert diese auf.

Das *Lernen* läuft über das Ändern der Gewichte und Biases der Neuronen ab. Das Ziel des Lernprozesses ist, den Ausgabe-Wert der Loss-Funktion zu minimieren. Hierfür verwendet man Gradient Descent. Diese Methode nutzt den Gradienten der Loss-Funktion an ihrem aktuellen Punkt, um ein lokales Minimum zu finden. Der Gradient gibt die

höchste Steigung an einem Punkt an, also die Veränderung an den Gewichten, die die Loss-Funktion maximieren würde. Nimmt man nun die umgekehrte Richtung, minimiert man sie. Man wendet Gradient Descent also immer wieder auf die Loss-Funktion an und passt die Gewichte entsprechend an.

Dieser Lernprozess wendet das Prinzip der *Backpropagation* an. Dieser Algorithmus geht die Layer von hinten (also vom Output-Layer) nach vorne (Input-Layer) durch und passt die Gewichte mithilfe von Gradient Descent an. Ist der Algorithmus beim Input-Layer angekommen, wird das nächste Datum optimiert, da in einem Schritt immer nur ein Datum betrachtet werden kann. Um die finalen Gewichte des Netzwerks zu erhalten, werden die für die einzelnen Werte ermittelten Gewichte gemittelt.

Abschließend wird das Neuronale Netzwerk noch mithilfe des Testdatensatzes getestet, dies soll Overfitting ausschließen. Overfitting ist ein Problem, das auftreten kann, wenn ein Netzwerk zu genau auf einen bestimmten Datensatz trainiert wird, dieser kann dann mit einer sehr hohen Genauigkeit erkannt werden, aber neue Daten werden falsch klassifiziert. Das Neuronale Netzwerk reagiert dann meist auf Hintergrundrauschen in den Daten, welches nichts mit den eigentlich wichtigen Werten zu tun hat. (vgl. [4])

# 3 Methoden - Erdbeben

## 3.1 Entstehung

Erdbeben entstehen aufgrund von Spannungen in der Erdkruste, welche über einen kurzen Zeitraum aufgelöst werden, durch dies wird Energie freigesetzt. Dies tritt hauptsächlich an Grenzen von tektonischen Platten auf, da diese konstant in Bewegung sind, wodurch sie auf verschiedene Arten belastet werden. Entweder baut sich hierbei Spannung auf, wenn zwei Platten aufeinander treffen und sich zusammendrücken. Alternativ ist es auch möglich, durch Zug auf eine Platte Spannung aufzubauen [5]. Zuletzt bleibt noch der Fall, dass zwei Platten sich aneinander vorbei bewegen und sich dabei verkanten und darüber Spannung aufbauen (Shear Stress) (vgl. [10]).

Eine Vorhersage von Erdbeben ist bisher nicht möglich, es gib allerdings Frühwarnsysteme, welche Erdbeben frühzeitig erkennen können. Generell lohnt sich dies aber hauptsächlich in Gegenden, in denen bereits ein Risiko für Erdbeben besteht.

## 3.2 Nachbeben

Meistens löst ein Erdbeben schwächere Nachbeben aus, da die aufgebaute Spannung nicht vollständig gelöst wird. Nachbeben können zwar auch nur bedingt vorher gesagt werden, aber es gibt einige Gesetzmäßigkeiten, mit denen sich Stärke und Anzahl an Nachbeben bestimmen lassen. (vgl. [9])

Die Anzahl Nachbeben wird durch *Omori's Law* beschrieben:

$$n(t) = \frac{k}{(c + t)^p}$$

$k, c$ : Konstanten abhängig vom Hauptbeben

$p$ : Abfallrate (normalerweise 0.7-1.5)

$t$ : Zeit

Die Stärke des Stärksten Nachbebens wird durch *Båth's Law* angegeben. Dieses besagt, dass das Verhältnis aus Stärke des stärksten Nachbebens und Stärke des Hauptbebens näherungsweise konstant ist. Meist mit einem Faktor zwischen 1,1 und 1,2. (vgl. [12])

# 4 Vorhersage von Nachbeben

## 4.1 Klassische Methoden

Die Vorhersage von Nachbeben ist mit verschiedenen Methoden möglich, die ohne Deep Learning Methoden auskommen, drei davon werden von DeVies et. al [3] aufgezeigt, eine weitere von Broccardo und Mignan [7].

Die Gebiete, für die eine Vorhersage von Nachbeben erfolgen soll, werden in Zellen von 5km x 5km x 5km aufgeteilt, welche aus einem Umkreis von 100km in der horizontalen und 50km in der vertikalen Richtung des Hauptbebens stammen.

Mit welcher Genauigkeit eine Methode Nachbeben vorhersagt ergibt sich aus der Fläche unter einer Kurve, die die Falsch-Positiv-Rate gegen die Richtig-Positiv-Rate plottet. Hierbei ist eine möglichst große Fläche gleichbedeutend mit einer hohen Genauigkeit. Die folgenden Kriterien können eine Wahrscheinlichkeit für ein Nachbeben in einer bestimmten Zelle angeben.

- Coulomb Failure Stress Change ( $\Delta\text{CFS}$ ) [3]
- Maximum Change in Shear Stress ( $\Delta\tau_{max}$ ) [3]
- von-Mises-Yield-Criterion ( $\sqrt{3\Delta J_2}$ ) [3]
- Abstand und Durchschnittliche Verschiebung (average Slip)[7]

Coulomb Failure Stress Change verwendet verschiedene Belastungsarten, aus denen sich dann eine Wahrscheinlichkeit ergibt, dass ein Nachbeben auftritt. Die Gesamtgenauigkeit ist bei dieser Methode nicht besonders hoch, sie liegt zwischen 50% und 60% (vgl. [3]). Beispiele für die Vorhersagen können in Abb. 4.1(a-d) und abb. 4.2(a,e) gesehen werden.  $\Delta\tau_{max}$  betrachtet die größte Veränderung in Scher-Belastung in einer gegebenen Zelle für die Vorhersage. Diese Methode ist um einiges genauer als  $\Delta\text{CFS}$  und erreicht für ein simuliertes Erdbeben 84,74% Genauigkeit.(Abb. 4.2(b,f))

Das von-Mises-Yield-Criterion verwendet die 2. Invariante des stress-change-Tensors. Auch diese Methode ist genauer als  $\Delta\text{CFS}$  und erreicht sehr ähnliche Werte wie  $\Delta\tau_{max}$ . Die Genauigkeit liegt für das simulierte Erdbeben bei 84,58%. (Abb. 4.2(c,g))

Die letzte Methode, welche von Broccardo und Mignan entwickelt wurde [7], verwendet eine Funktion abhängig von der in einer Zelle auftretenden Verschiebung in Metern  $d$  und dem minimalen Abstand zum Hauptbeben  $r$  (auch in Metern). Außerdem verwendet die entwickelte Formel drei durch Regression bestimmte Konstanten  $\beta_0, \beta_1$  und  $\beta_2$ . Die Formel sieht wie folgt aus:

$$Pr(y) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \log r + \beta_2 \log d)}}$$



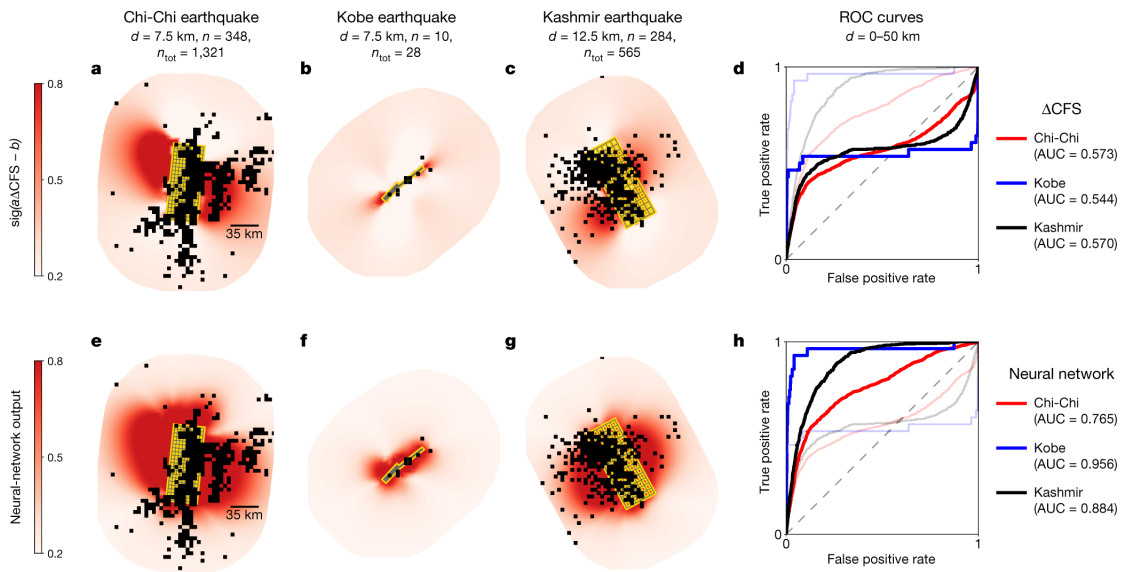


Abbildung 4.1: Vergleich des Deep Learning Ansatzes zur Realität [3]

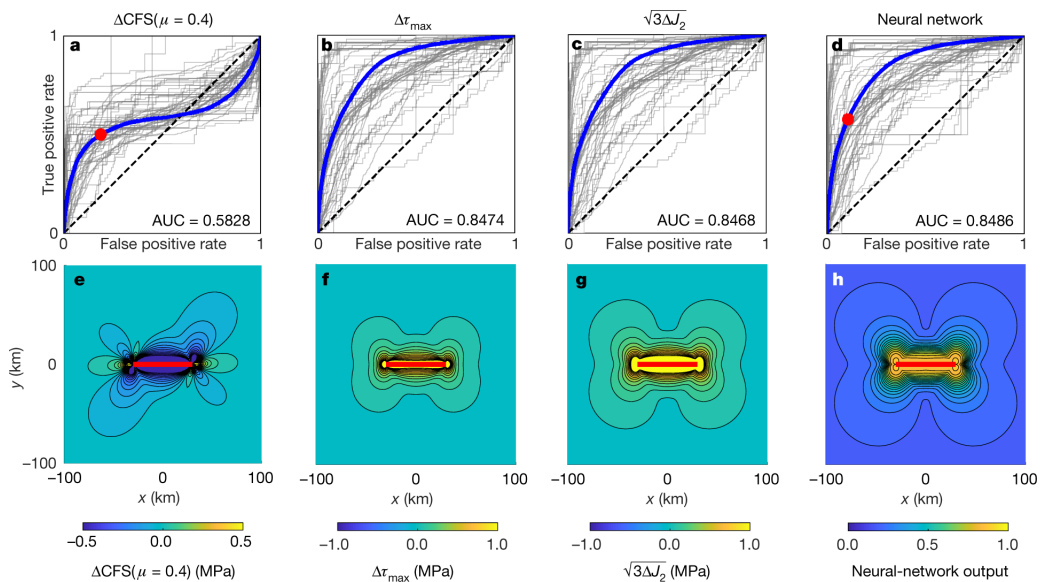


Abbildung 4.2: Vergleich des Deep Learning Ansatzes zu herkömmlichen Methoden [3]

Diese Methode erreicht eine Genauigkeit von 86,1%, eine Beispiel Vorhersage kann in Abb. 4.3 gesehen werden.

## 4.2 Deep Learning Ansatz

Das Neuronale Netzwerk von DeVries et. al [3] verwendet 6 Hidden Layers mit je 50 Neuronen. Das Netzwerk ist zwischen den Layern vollständig verknüpft und hat 13.451 Gewichte und Biases, die im Training verändert werden können. Als Output-Layer dient

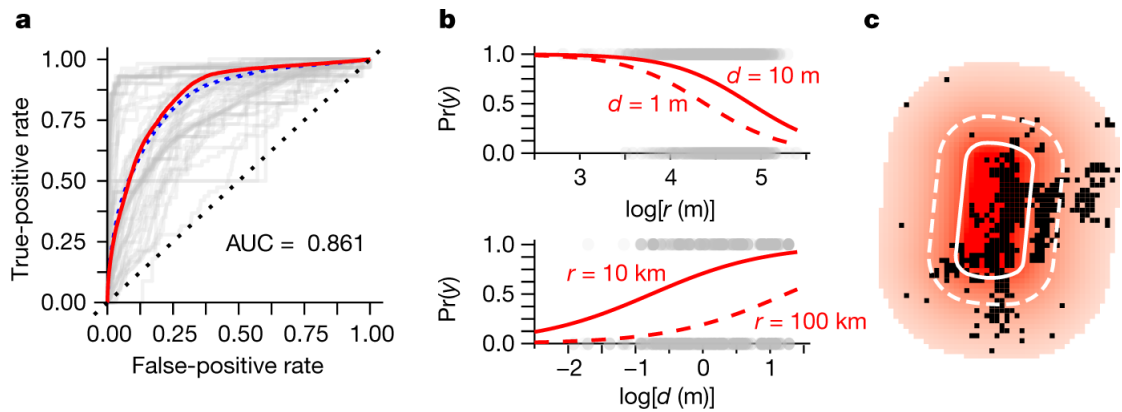


Abbildung 4.3: Genauigkeit der von Broccardo und Mignan entwickelten Methode[7]

ein einziges Neuron, dessen Aktivierung der Wahrscheinlichkeit eines Nachbebens in einer Zelle entspricht. Für den Input-Layer werden die Beträge von 6 unabhängigen Komponenten des stress-change-Tensors aus dem Mittelpunkt der jeweiligen Zelle verwendet. Dabei verwendet das Netzwerk eine Tangenshyperbolicus-Funktion (Abb. 2.2c) als Aktivierungsfunktion.

Für das Training wurden 75% der Daten verwendet, der Rest blieb zum Testen übrig. Die Daten wiesen jeder Zelle eine 1 oder eine 0 zu, je nachdem, ob ein Nachbeben auftrat oder nicht.

In Abbildung 4.1(e-h) kann die Genauigkeit des Neuronalen Netzwerks im Vergleich zu  $\Delta\text{CFS}$  anhand von realen Ereignissen gesehen werden. In Abbildung 4.2(d,h) wird ein simuliertes Erdbeben verwendet. Das Netzwerk erzielt hierbei eine Genauigkeit zwischen 76,5% und 95,6%, wobei der Wert beim simulierten Erdbeben bei 84,86% liegt.

## 5 Zusammenfassung

Im Vergleich der Methoden zeichnet sich deutlich ab, dass die Vorhersage von Nachbeben durch Deep Learning um einiges genauer ist, als die Vorhersage mit dem Coulomb-Failure-Stress-Change. Aber es ist auch zu betrachten, dass andere Methoden wie  $\Delta\tau_{max}$ , das von-Mises-Yield-Criterion oder der Abstand zum Hauptbeben und die Verschiebung ähnlich gute Ergebnisse erzielen.

Deep Learning hat also gegenüber klassischen Methoden keinen Vorteil bezüglich Genauigkeit.

Das Problem eines Neuronalen Netzwerks ist die Interpretierbarkeit, welche gerade für eine Anwendung in der Wissenschaft zentral ist. Hier ist es nötig, dass genau nachvollzogen werden und nötigenfalls die Korrektheit eines Vorgehens bewiesen werden kann. Gerade bei noch größeren Netzwerken als dem von DeVries et. al ist dies dann nur noch sehr schwer bis gar nicht möglich.

Auf der anderen Seite ist ein Neuronales Netzwerk gegebenenfalls günstiger und weniger rechenaufwendig, als eine detaillierte Simulation der physikalischen Gegebenheiten. Aufwendig ist hier das Training, das Anwenden kann auch mit weniger Rechenkapazitäten erfolgen.

Allerdings sind auch die klassischen Methoden welche von deVries et. al und auch die von Broccardo und Mignan entwickelte Methode nicht sehr rechenaufwendig.

In einem Anwendungsgebiet, in dem klassische Methoden insgesamt ungenauer sind (vergleichbar mit dem Coulomb-Failure-Stress-Change) und ansonsten nur teure Simulationen genauere Ergebnisse liefern, könnte ein Neuronales Netzwerk von Nutzen sein.

Das Problem der Interpretierbarkeit bleibt zwar bestehen, je nachdem, was erreicht werden soll ist dies aber vernachlässigbar. Ist zum Beispiel das Ziel nur die genaue Vorhersage von Nachbeben oder generell Erdbeben, dann ist es nützlicher, in der Lage zu sein, diese frühzeitig zu erkennen und Menschen rechtzeitig zu warnen. Der Nutzen darin, zu verstehen, warum ein Nachbeben aufgetreten ist, hat für dieses dann weniger Bedeutung.

Zusammenfassend lässt sich also sagen, dass ein Nutzen für Deep Learning in der Wissenschaft durchaus besteht, aber stark davon abhängt, wie genau eine Künstliche Intelligenz verwendet wird und wie verständlich das Modell letztendlich ist, um tatsächlich Erkenntnisse aus ihm ziehen zu können.

# Literaturverzeichnis

- [1] Neuronales netz abbildung. <https://www.researchgate.net/profile/Facundo-Bre/publication/321259051/figure/fig1/AS:614329250496529@1523478915726/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o.png> (aufgerufen am 26.08.2021).
- [2] CodeBullet. What are neural networks || how ais think. <https://www.youtube.com/watch?v=JeVDjExBf7Y>, (aufgerufen am 06.04.2020), 2018.
- [3] P. M. R. DeVries, F. Viégas, M. Wattenberg, and B. J. Meade. Deep learning of aftershock patterns following large earthquakes. *Nature*, 560:632–634, 8 2018.
- [4] T. Dietterich. Overfitting and undercomputing in machine learning. *ACM Comput. Surv.*, 27(3):326–327, Sept. 1995.
- [5] ESKP. Wie entstehen erdbeben. <https://www.eskp.de/grundlagen/naturgefahren/wie-entstehen-erdbeben-935103/> (aufgerufen am 15.06.2021).
- [6] D. D. Janssen. Machine learning in der schule. <https://www.science-on-stage.de/sites/default/files/material/machine-learning-in-der-schule.pdf>, (aufgerufen am 02.05.2020), 2020.
- [7] A. Mignan and M. Broccardo. One neuron versus deep leraning in aftershock prediction. *Nature*, 574:E1–E3, 10 2019.
- [8] G. Sanderson. Neural networks. [https://youtube.com/playlist?list=PLZHQ0b0WTQDNU6R1\\_67000Dx\\_ZCJB-3pi](https://youtube.com/playlist?list=PLZHQ0b0WTQDNU6R1_67000Dx_ZCJB-3pi), 2018.
- [9] Spektrum. Nachbeben. <https://www.spektrum.de/lexikon/geowissenschaften/nachbeben/10891> (aufgerufen am 15.06.2021).
- [10] USGS. Definition of shear stress. <https://earthquake.usgs.gov/learn/glossary/?term=shear%20stress>, (augerufen am 11.06.2021).
- [11] USGS. Definition of slip. <https://earthquake.usgs.gov/learn/glossary/?term=slip>, (augerufen am 09.06.2021).
- [12] Wikipedia. Aftershock. <https://en.wikipedia.org/wiki/Aftershock>, (abgerufen am 11.06.2021).
- [13] Wikipedia. Tensor. <https://de.wikipedia.org/wiki/Tensor>, (abgerufen am 07.06.2021).