



BACHELORARBEIT

für die Prüfung zum

BACHELOR OF SCIENCE

des Studiengangs Informationstechnik

der Dualen Hochschule Baden-Württemberg Mannheim

THEMA

Prognose von Retirement Curves von Flugzeugen mithilfe Neuronaler Netze

von

Clemens Schlesinger

9. September 2020

Bearbeitungszeitraum:	24.06.2020 – 13.09.2020
Matrikelnummer, Kurs:	9566062, TINF17ITIN
Abteilung:	Institut für Flughafenwesen und Luftverkehr
Ausbildungsfirma:	Deutsches Zentrum für Luft- und Raumfahrt e. V.
Betreuer der Ausbildungsfirma:	M.Sc. in CS Martin Jung

Ehrenwörtliche Erklärung

Gemäß §5 (3) der Studien- und Prüfungsordnung DHBW Technik vom 22. September 2011.

Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Köln, den 9. September 2020

Zusammenfassung

Das Ziel dieser Arbeit ist die Entwicklung eines Neuronalen Netzes, das die Überlebenswahrscheinlichkeit von Flugzeugen vorhersagen kann.

Hierzu werden, für ein besseres Verständnis von Künstlicher Intelligenz, die drei Arten des Machine Learning untersucht, sowie Neuronale Netze detailliert erklärt. Für die Entwicklung eines Programms zur Berechnung und Visualisierung von Retirement Curves, werden die Python Bibliotheken Keras und TensorFlow verwendet. Die Trainingsdaten für das Neuronale Netz sind die Daten zur weltweiten Passagierflugzeugflotte, welche mit Wirtschaftsdaten zum weltweiten Bruttoinlandsprodukt pro Kopf, dem Ölpreis und den Flugpassagierzahlen pro Jahr angereichert werden. Die Datenauswahl wird detailliert geschildert und es wird die Bewältigung von Herausforderungen bei der Einbindung der Wirtschaftsdaten festgehalten. Mithilfe des Kolmogorov-Smirnov-Tests und des Vergleichs mit bereits existierenden Vorhersagen werden die erstellten Retirement Curves validiert.

Die Entwicklung des Programms konnte unter Erfüllung aller Anforderungen vollständig abgeschlossen werden.

Abstract

The goal of this thesis is to develop a neural network to predict aircraft survival probability.

For a better understanding of artificial intelligence, three types of machine learning are examined and neural networks are explained in detail.

In terms of the development of a program calculating and visualizing Retirement Curves the Python libraries Keras and TensorFlow are applied.

As training data for the neural network those on worldwide passenger aircraft fleet are complemented with economic data on worldwide gross domestic product per capita, oil price and the number of air passengers per year.

The process of data selection is a matter of discussion as well as the challenges of integrating economic data.

Finally, using the Kolmogorov-Smirnov test and in comparison with already existing predictions, the created Retirement Curves are validated.

After all, the development was completed, while fulfilling all requirements.

Inhaltsverzeichnis

Abbildungsverzeichnis	VI
Tabellenverzeichnis	VIII
Abkürzungsverzeichnis	IX
1 Einleitung	1
2 Motivation	3
3 Related Work	6
4 Grundlagen	8
4.1 Retirement Curves	8
4.2 Drei Arten von Machine Learning	10
4.2.1 Überwachtes Lernen	11
4.2.2 Unüberwachtes Lernen	12
4.2.3 Verstärkendes Lernen	13
4.3 Neuronale Netze	14
4.3.1 Herkunft der Idee der Neuronalen Netze	14
4.3.2 Aufbau eines Neurons	15
4.3.3 Aufbau eines Neuronalen Netzes	17
4.3.4 Backpropagation	20
4.4 TensorFlow und Keras	21
5 Datenauswahl	24
5.1 Art der Vorhersage	24

5.2	Trainingsdaten	25
5.2.1	Vermutete Auswirkungen der Eingabeparameter	25
5.2.2	Signifikanz der Eingabeparameter	30
5.3	Vorbereitung der Trainingsdaten	31
5.4	Schwierigkeiten bei der Datenauswahl	32
5.4.1	Einbinden von Wirtschaftsdaten	32
5.4.2	Wahl zwischen festen Werten und Veränderungsraten	32
5.5	Entwicklung von Szenarien	33
6	Analyse und Konzeption	35
6.1	Anforderungsanalyse	35
6.2	Programmstruktur	35
6.3	Spezifikationen	37
7	Umsetzung	39
7.1	Training der Neuronalen Netze	39
7.1.1	Trainingsdaten vorbereiten	39
7.1.2	Optimieren der Netze	40
7.1.3	Spezifikationen der Neuronalen Netze	45
7.1.4	Speichern der Netze	46
7.2	Implementierung der Vorhersage	46
8	Validierung	49
8.1	Vergleich mit den Trainingsdaten	49
8.2	Vergleich mit anderen Vorhersagemodellen	52
8.3	Scheinkorrelation durch Inflation	55
9	Ergebnisse	56
10	Ausblick	59
	Literatur	61

Abbildungsverzeichnis

2.1	Flugzeugbestellungen bei Boeing zwischen 2003 und 2019	4
3.1	Google Trends - Künstliche Intelligenz [26]	6
4.1	Retirement Curves	9
4.2	Drei Arten von Machine Learning [43]	10
4.3	Unterscheidung zwischen Blumen anhand ihrer Blütenblätter [28] . .	11
4.4	Clustering [11]	12
4.5	Anomalieerkennung [11]	12
4.6	Agentenbasiertes Lernen [11]	13
4.7	Der Schachtürke [48]	14
4.8	Arbeitsspeicher Preise zwischen 1997 und 2011 [4]	15
4.9	Aufbau eines Neurons	16
4.10	Sigmoid und ReLU Aktivierungsfunktion [1]	17
4.11	Zeichnung des Mittelhirns eines Spatzen [39]	18
4.12	Struktur eines KNN [51]	19
4.13	Stochastic Gradient Descent vs Gradient Descent [41]	20
5.1	Trainingsdaten visualisiert (Widebody)	27
5.2	Trainingsdaten visualisiert (Turboprop)	27
5.3	Entwicklung des GDP	28
5.4	Entwicklung des Ölpreises	29
5.5	Entwicklung der Passagierzahlen	30
6.1	Programmstruktur	36
6.2	Gewünschte Ausgabe des Neuronales Netzes	38

7.1	Vorhersage für Narrowbody Flugzeuge	43
7.2	Vorhersage für Widebody Flugzeuge	43
8.1	Flugzeuge des Typs Boeing 747 in den Trainingsdaten	51
8.2	Vorhergesagte Überlebenswahrscheinlichkeit der Boeing 747	51
8.3	Vergleich von Narrowbody Trainingsdaten und Vorhersage	52
8.4	Retirement Curve Prognose von FW	53
8.5	Retirement Curve Prognose des Neuronalen Netzes	54
8.6	Visualisierung der Trainingsdaten	54
9.1	Ausgabe des Programms	57
9.2	Vergleich verschiedener Wirtschaftsszenarien	58

Tabellenverzeichnis

5.1	Vermutete Auswirkungen der verwendeten Merkmale	26
5.2	Parameterkombinationen für die Signifikanzanalyse	30
5.3	Kombination der Szenarien	33
6.1	Format der Eingabedaten	37
7.1	Ersetzen von Entwicklungsmodellen durch reale Werte	39
7.2	Formen des Neuronalen Netzes	41
7.3	Testdaten für die Optimierung	42
7.4	Kombination der Kompilierungsparameter	44
8.1	Kolmogorov-Smirnov-Test für Narrowbody Prognose	50
8.2	Kolmogorov-Smirnov-Test: Vergleich Prognose FW und NN	53
9.1	Drei Wirtschaftsszenarien für den Vergleich	57

Abkürzungsverzeichnis

NN Neuronales Netz

DLR Deutsches Zentrum für Luft- und Raumfahrt e. V.

FW Institut für Flughafenwesen und Luftverkehr

KI Künstliche Intelligenz

CAEP Committee on Aviation Environmental Protection

ICAO International Civil Aviation Organization

VBA Visual Basic for Applications

NNN Natürliches Neuronales Netz

KNN Künstliches Neuronales Netz

ReLU Rectified Linear Unit

SGD Stochastic Gradient Descent

GDP Gross Domestic Product

IEA International Energy Agency

HDF5 Hierarchical Data Format

PNG Portable Network Graphic

JPEG Joint Photographic Experts Group Standard

1 Einleitung

In der aktuellen Corona-Situation und der dadurch bedingten Rezession vieler Fluggesellschaften fallen kostensparende Maßnahmen besonders ins Gewicht. Dadurch wird die Flottenplanung der Unternehmen wichtiger als je zuvor, um, durch ein möglichst langes Nutzen der Flugzeuge und rechtzeitige Neuanschaffungen, die anfallenden Kosten so gering wie möglich zu halten.

Die Vorhersage von Retirement Curves spielt eine wichtige Rolle bei der Planung von Flugzeugflotten. Zu wissen, wie wahrscheinlich ein Flugzeug noch weitere zehn Jahre fliegen kann, ist von großer Bedeutung, wenn es darum geht, die Flotten für die nächsten Jahre zu planen, damit rechtzeitig neue Flugzeuge bestellt werden können und das allgemeine Verständnis des Flugzeullebenszyklus verbessert wird.

Es gibt diverse Artikel, die sich mit der Flottenplanung von Fluggesellschaften befassen ([10], [50], [9]), jedoch wählt keine dieser Arbeiten ein Neuronales Netz (NN) als Ansatz.

In dieser Arbeit wird ein Neuronales Netz für die Vorhersage von Retirements von Flugzeugen entwickelt und trainiert. Dabei werden Trainingsdaten für das Neuronale Netz ausgewählt und anhand ihrer Signifikanz bewertet. Es wird ein Programm entwickelt, das anhand des Neuronalen Netzes eine Vorhersage für eingegebene Daten erstellt. Ein besonderer Fokus wird dabei auf die Optimierung des Neuronalen Netzes sowie die Form der Eingabedaten gelegt.

Kapitel 2 befasst sich mit der Motivation zu dieser Arbeit. In Kapitel 3 werden Problemstellungen beschrieben, die denen der Aufgabenstellung ähnlich sind. Ebenso wird die Aktualität des Problems dargestellt. Kapitel 4 beschäftigt sich mit den Grundlagen dieser wissenschaftlichen Arbeit. Dazu gehören Retirement Curves und

Ansätze von Machine Learning, sowie Neuronale Netze. In Kapitel 5 wird die Auswahl der Trainingsdaten für ein Neuronales Netz beschrieben und diese wird in Kapitel 8 validiert. Anhand einer Anforderungsanalyse wird in Kapitel 6 eine Programmstruktur konzeptioniert. Basierend auf dieser wird die Umsetzung des Programms in Kapitel 7 dargestellt. Abschließend werden die Ergebnisse dieser Arbeit in Kapitel 9 vorgestellt und zuletzt erfolgt in Kapitel 10 ein Ausblick für zukünftige Arbeiten zum Thema.

2 Motivation

Im Institut für Flughafenwesen und Luftverkehr (FW) des Deutschen Zentrum für Luft- und Raumfahrt e. V. (DLR) werden Retirement Curves (siehe Abschnitt 4.1) von Flugzeugen per Hand berechnet. Die Retirement Curves werden benötigt, um die Entwicklung der Zusammensetzung der globalen Flugzeugflotte prognostizieren zu können.

Der Ablauf der Prognose ist dabei wie folgt:

- Verkehrsprognose: Flugbewegungen und Streckenpassagiere pro Flughafenpaar
- Prognose Sitzladefaktor [3]
- Angebotene Sitzplätze pro Flughafenpaar ermitteln
- **Retirement Curve:** Bestimmung der Wahrscheinlichkeit, dass ein bestimmtes Flugzeug im Prognosejahr noch in Betrieb ist
- Anwendung auf die gesamte Flotte aus dem Basisjahr → Bestimmung der überlebenden Flugzeuge im Prognosejahr
- Anwendung des Reduktionsfaktors der Flotte auf die geflogenen Frequenzen auf jedem Flughafenpaar
- Berechnung pro Flughafenpaar: Wie viele Sitzplätze/Flüge werden laut Prognose angeboten - Wie viele werden durch Flugzeuge aus dem Basisjahr erbracht
- Hinzufügen der passenden Flugzeuge, um Differenz „Benötigte Flüge minus Flüge mit Flugzeugen aus dem Basisjahr“ und „Benötigte Sitzplätze minus Sitzplätze auf Flügen mit Flugzeugen aus dem Basisjahr“ zu decken

- Bestimmung wie viele Flugzeuge benötigt werden, um die hinzugefügten Flüge durchzuführen

Die Retirement Curves werden als Teil eines Prognoseprozesses verwendet. Da sie bisher per Hand als Regression bestimmt werden [7, 54], soll die Ermittlung von Retirement Curves mithilfe Neuronaler Netze eine deutliche Effizienzsteigerung für die Prognose der Zusammensetzung der globalen Flugzeugflotte erwirken.

Die Prognose der Retirement Curves wird nicht nur von Modellierern für das Erstellen von Entwicklungsmodellen, sondern auch von Flugzeugherstellern verwendet, da diese die Werte für die Prognose der verkauften Flugzeuge benötigen.

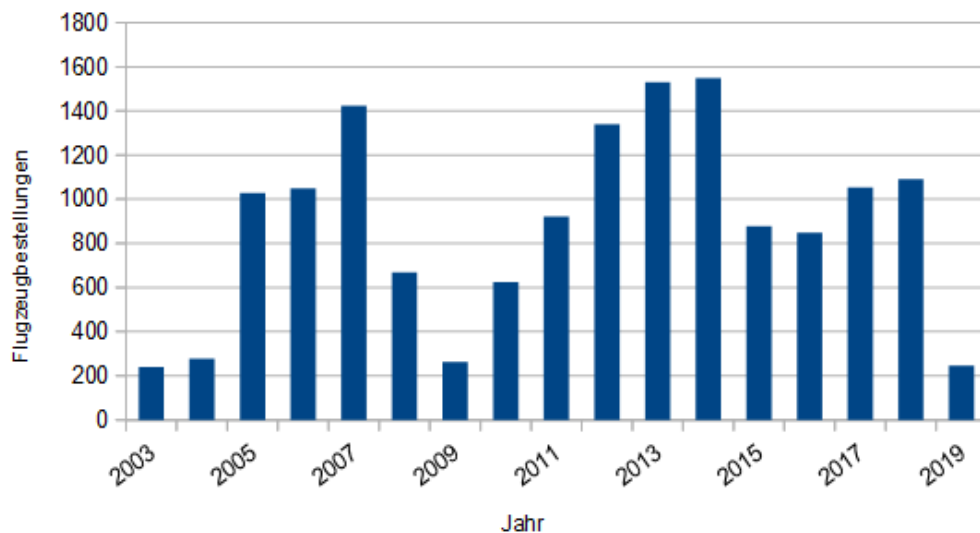


Abbildung 2.1: Flugzeugbestellungen bei Boeing zwischen 2003 und 2019

Das Unternehmen Boeing veröffentlicht auf seiner Website die Verkaufszahlen der letzten Jahre (Abbildung 2.1). [31]

Es ist zu erkennen, dass die Bestellmengen von Jahr zu Jahr stark variieren, da sie von unterschiedlichen wirtschaftlichen Faktoren wie Ölpreis und Kaufkraft, aber auch der Anzahl an ausgemusterten Flugzeugen abhängen. Damit für die Zukunft die richtigen Größenordnungen der Bestellmenge von Flugzeugen eingeschätzt werden können, sind die zu ermittelnden Retirement Curves für Flugzeugbauer wie Boeing und Airbus von großer Bedeutung.

Neuronale Netze können in Daten für Menschen nicht sichtbare Zusammenhänge erkennen. Daher ist ein Neuronales Netz, in welches ausreichend Trainingsdaten eingespeist werden, für die Berechnung von Retirement Curves sehr gut geeignet. Mithilfe von TensorFlow und Keras (siehe Abschnitt 4.4) kann ein NN ausgearbeitet und optimiert werden, um aus den ausgewählten Trainingsdaten (Kapitel 5) die Überlebenswahrscheinlichkeit von Flugzeugen zu ermitteln.

Die Wahl dieses neuen Ansatzes bietet ein gutes Erweiterungspotential, da bei Machine Learning durch Anpassen der Trainingsdaten sehr gut skaliert werden kann. Darüber hinaus ist die Verwendung eines Neuronalen Netzes zukunftsorientiert und wenn das Modell gut trainiert wird, können in den nächsten Jahrzehnten Retirement Curves damit vorhergesagt werden.

3 Related Work

Seit fünf Jahren steigt das Interesse an Künstlicher Intelligenz (KI) und Maschinellern Lernen stetig an. Abbildung 3.1 zeigt die Google Suchanfragen zu Themen in diesem Bereich. Zu erkennen ist, dass das Interesse allgemein gestiegen ist.

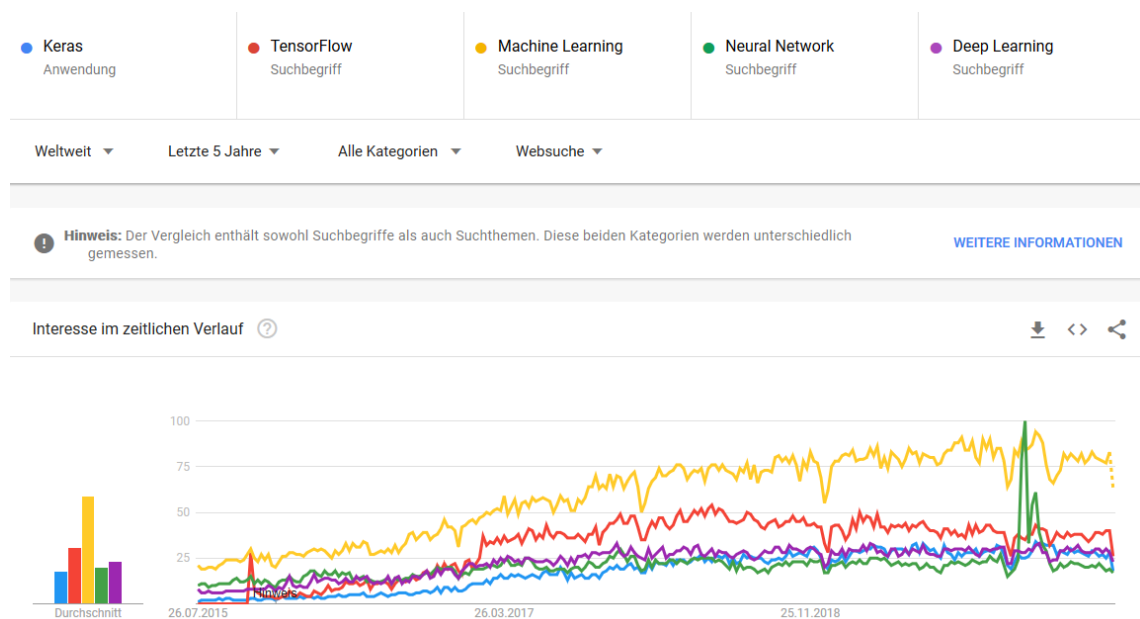


Abbildung 3.1: Google Trends - Künstliche Intelligenz [26]

Dieser ansteigenden Aufmerksamkeit folgend, gibt es auch diverse Arbeiten zum Thema Predictive Maintenance. Hashemian [13], Wetzler [52] und Grall u. a. [12] verwenden noch manuelle Ansätze zur Berechnung der Predictive Maintenance, wohingegen Susto u. a. [42], Heng u. a. [16] und Langone u. a. [27] schon Ansätze mit KI verwenden. Dabei werden bereits gute Ergebnisse in der Vorhersage erreicht.

Predictive Maintenance befasst sich im Allgemeinen mit der Vorhersage von Wartungsarbeiten, Fehlern oder Ähnlichem in der Industrie. Das Ziel ist die proaktive Wartung von Maschinen, um die Störzeiten zu minimieren. [32]

Mit der Vorhersage von Retirement Curves befassen sich vor allem das Committee on Aviation Environmental Protection (CAEP) der Internationalen Zivilluftfahrtorganisation (International Civil Aviation Organization, ICAO) und das Institut für Flughafenwesen und Luftverkehr des DLR.

Bisher wird die Vorhersage der Retirement Curves größtenteils per Hand durchgeführt [5, 17, 20], wobei bei FW neben der händischen Analyse [38] auch schon erste Ansätze mit Neuronalen Netzen getestet werden.

Für die händische Vorhersage werden Formeln entwickelt, die aufgrund der Eingabeparameter wie Wirtschafts- und Flugzeugdaten eine Ausgabe bestimmen. In ihrer Veröffentlichung beschreiben Chao u. a. [6] die Entscheidungen bei der Planung von Flugzeugflotten. Dafür werden drei Szenarien erstellt, welche die Entwicklung der Passagiernachfrage beschreiben. Anhand der unterschiedlichen Ausgangsdaten werden für die drei Szenarien die Entwicklung der Flugzeugflotte und der Einfluss auf die Umwelt abgeglichen.

Schaefer [38] beschreibt in seinem Forschungsbericht die Ermittlung der Anzahl von ausgemusterten Flugzeugen. Dabei wird die Anzahl der Flugzeuge mithilfe von Visual Basic for Applications (VBA) berechnet. Es werden zwei unterschiedliche Ansätze beschrieben. Der erste Ansatz verfolgt eine lineare Projektion in die Zukunft, während bei dem zweiten Ansatz die existierenden Retirement Curves eins zu eins in die Zukunft übernommen werden.

4 Grundlagen

Im folgenden Kapitel wird ein Überblick von Retirement Curves gegeben. Außerdem wird ein kurzer Ausblick auf Machine Learning vorgenommen. Abschließend werden Neuronale Netze und die verwendeten Python Bibliotheken `TensorFlow` und `Keras` vorgestellt.

4.1 Retirement Curves

Die Analyse der Überlebenswahrscheinlichkeit von Maschinen und Fahrzeugen ist ein Teil der Predictive Maintenance Forschung. Um die kürzest mögliche Ausfallzeit von wichtigen Komponenten zu erlangen, wird in der Predictive Maintenance anhand verschiedener Parameter die Wahrscheinlichkeit für eine Wartung oder einen Ausfall ermittelt. Auch in der Medizin findet Predictive Maintenance Anwendung, da bei wichtigen Operationen alle Geräte einsatzbereit sein müssen. Ein Mittel der Darstellung ist das Erstellen eines Graphen, in dem Überlebenswahrscheinlichkeiten in Abhängigkeit zum jeweiligen Alter aufgetragen werden. Diese Darstellungsform nennt sich „Retirement Curve“.

Diese Retirement Curves werden auch für die Vorhersage der maximalen Lebensdauer von Flugzeugen verwendet. Hierbei sind es Graphen, auf denen die Wahrscheinlichkeit, dass ein Flugzeug noch flugfähig, also noch im Bestand der Luftflotte ist, in Abhängigkeit vom Alter aufgetragen wird. Dabei spielen viele Faktoren, unter anderem der Flugzeugtyp, eine maßgebliche Rolle. Eine genaue Analyse der Faktoren erfolgt in Kapitel 5.

4.1 Retirement Curves

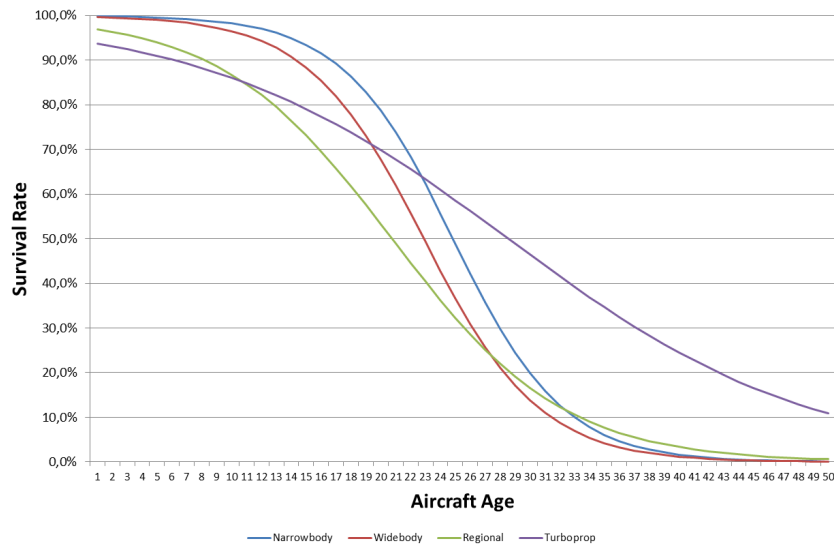


Abbildung 4.1: Retirement Curves

Abbildung 4.1 zeigt Retirement Curves von Flugzeugen, die bei FW entwickelt wurden. Es ist zu erkennen, dass unterschiedliche Flugzeugtypen sich in ihrer Lebenserwartung durchaus stark unterscheiden. Auf der X-Achse ist das Alter der Flugzeuge aufgetragen. Abhängig vom Alter findet sich auf der Y-Achse die Überlebenswahrscheinlichkeit eines Flugzeugs dieses Alters. Die Überlebenswahrscheinlichkeit beschreibt die Wahrscheinlichkeit, dass ein Flugzeug des beschriebenen Alters und Typs noch nicht ausgemustert wurde. Die Wahrscheinlichkeit wird nicht für einzelne Flugzeuge separat berechnet, sondern für alle Flugzeuge eines bestimmten Typs pauschal ermittelt.

Da jedes Jahr neue Flugzeuge entwickelt werden und die Nachhaltigkeit bei der Entwicklung von Flugzeugen eine große Rolle spielt, verändert sich die Lebenserwartung von neuen Flugzeugen von Jahr zu Jahr. Auch die aktuelle Wirtschaftslage hat einen Einfluss auf die Neuanschaffung von Flugzeugen, die wiederum zur Ausmusterung von alten Flugzeugen führen. Deswegen werden die Retirement Curves jedes Jahr aktualisiert und mit ihrer Hilfe sollen Prognosen für den Flugzeugbedarf des nächsten Jahres erstellt werden. In dieser Arbeit wird jedoch versucht Retirement Curves für einen längeren Zeitraum zu erstellen, um für entsprechend längere Phasen Flugzeugbedarfe vorherzusagen.

4.2 Drei Arten von Machine Learning

Das Ziel von Machine Learning ist, dass der Computer aus vorbereiteten Daten lernt. Dabei gibt es verschiedene Herangehensweisen.

Machine Learning wird allgemein in drei unterschiedliche Arten eingeteilt. Es gibt überwachtes, unüberwachtes und verstärkendes Lernen. [35]

Abbildung 4.2 zeigt, dass diese unterschiedlichen Arten auf verschiedenen Anwendungsgebieten eingesetzt werden. Überwachtes Lernen wird zur Vorhersage von stetigen Werten sowie der Einteilung in Klassen verwendet. Unüberwachtes Lernen wird zu Gruppierung von unbekanntem Daten verwendet. Durch verstärkendes Lernen wird ein Agent, also ein zu trainierendes Objekt, so trainiert, dass er selbstständig eine Aufgabe erfüllen kann.

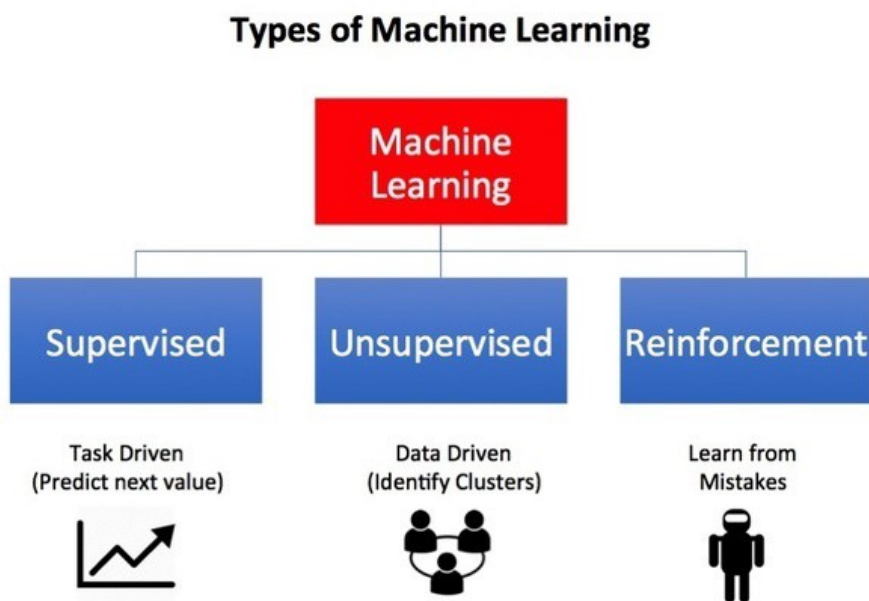


Abbildung 4.2: Drei Arten von Machine Learning [43]

In den folgenden Abschnitten wird ein kurzer Überblick über alle drei Arten vermittelt. Für die Vorhersage der Retirement Curves wird ein Neuronales Netz mit überwachtem Lernen genutzt. Daher wird überwachtes Lernen detaillierter vorgestellt, bevor auf die Neuronalen Netze eingegangen wird.

4.2.1 Überwachtes Lernen

Beim überwachten Lernen werden gekennzeichnete Trainingsdaten verwendet, um den Machine Learning Algorithmus zu trainieren. Gekennzeichnet bedeutet, dass die Trainingsdaten bereits klassifiziert sind. Der Algorithmus bestimmt dann für die gegebenen Merkmale (Eigenschaften der Daten) Gewichte, inwieweit dieses Merkmal in die Ermittlung der zugehörigen Klasse/des zugehörigen Wertes einfließt. Hierbei werden in mehreren Iterationen die Gewichte angepasst und anhand des daraus resultierenden Fehlers werden die Gewichte aktualisiert, bis eine für die Trainingsdaten optimale Vorhersage erreicht ist [36].

Abbildung 4.3 zeigt die Unterscheidung zwischen verschiedenen Blumen anhand der Größe ihrer Blütenblätter. Die Länge und Breite der Blütenblätter sind die Merkmale der Daten, mittels derer die Zugehörigkeit der Blume zu einer der drei Klassen bestimmt wird.

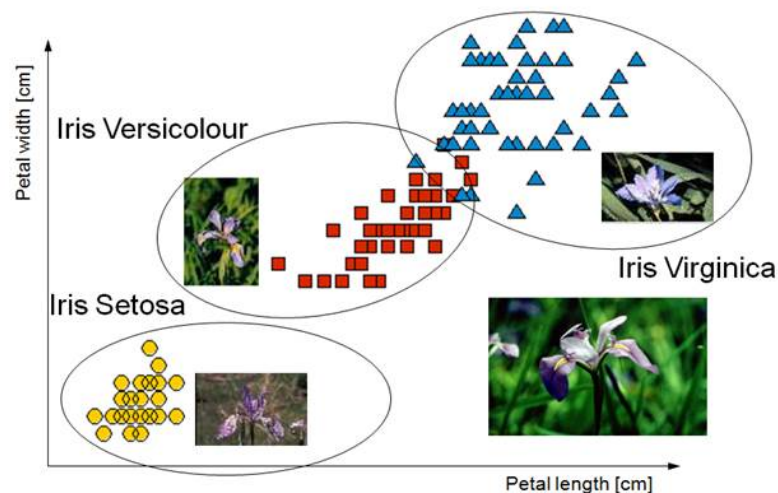


Abbildung 4.3: Unterscheidung zwischen Blumen anhand ihrer Blütenblätter [28]

Beim überwachten Lernen muss darauf geachtet werden, dass der Algorithmus nicht an die Trainingsdaten überangepasst ist. Beispielsweise ist ein Algorithmus, der bei den Trainingsdaten 100% und im laufenden Betrieb (mit unbekanntem Daten) nur 80% richtig vorhersagt schlechter, als ein Algorithmus, der im Training nur 90%,

dafür aber im laufenden Betrieb 85% erreicht. Beim Training muss also durch eine geschickte Wahl der Parameter Überanpassung verhindert werden.

4.2.2 Unüberwachtes Lernen

Beim unüberwachten Lernen werden nicht gekennzeichnete Daten verarbeitet. Beim sogenannten Clustering werden zusammenhängende Daten erkannt und gekennzeichnet (Abbildung 4.4). Des Weiteren können bei zusammengehörigen Daten Ausreißer erkannt und entfernt werden (Abbildung 4.5).

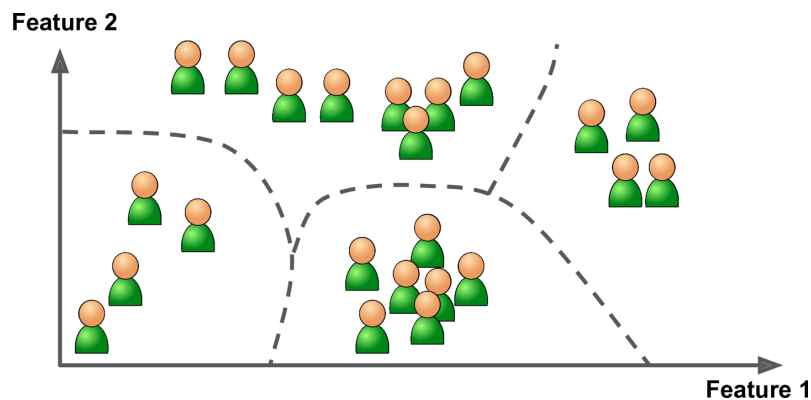


Abbildung 4.4: Clustering [11]

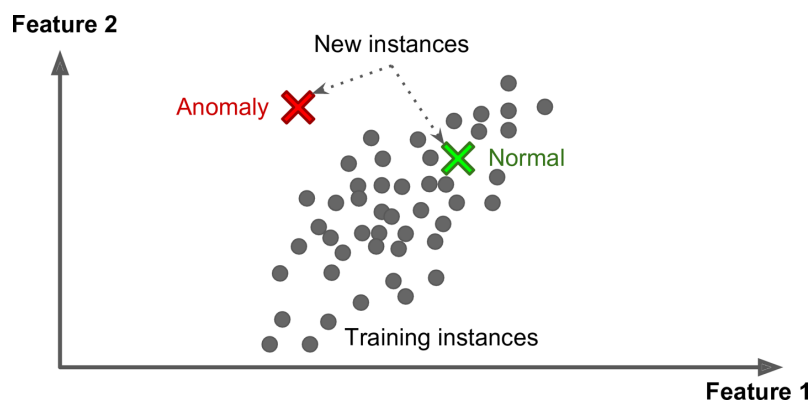


Abbildung 4.5: Anomalieerkennung [11]

Für Clustering muss der Algorithmus in den Merkmalen der Daten Zusammenhänge und Unterschiede erkennen, um zusammengehörige Daten kennzeichnen zu können.

Da für die Vorhersage der Retirement Curves ein Neuronales Netz verwendet werden soll, wird hier nicht genauer auf unüberwachtes Lernen eingegangen.

4.2.3 Verstärkendes Lernen

Verstärkendes Lernen funktioniert agentenbasiert. Ein oder mehrere Agenten führen Aufgaben aus und lernen aus einer resultierenden Belohnung oder Bestrafung, ob der ausgeführte Schritt gut war. Basierend darauf werden dann Entscheidungen getroffen. Abbildung 4.6 zeigt einen Agenten der ein Feuer löschen soll. Da er sich für die falsche Aktion entscheidet, wird er bestraft und lernt daraus für die nächste Iteration.

Mit verstärkendem Lernen können viele Optimierungsprobleme bewältigt werden, da ein Optimum gefunden werden soll und der Agent trainiert wird, bis er dieses erreicht.

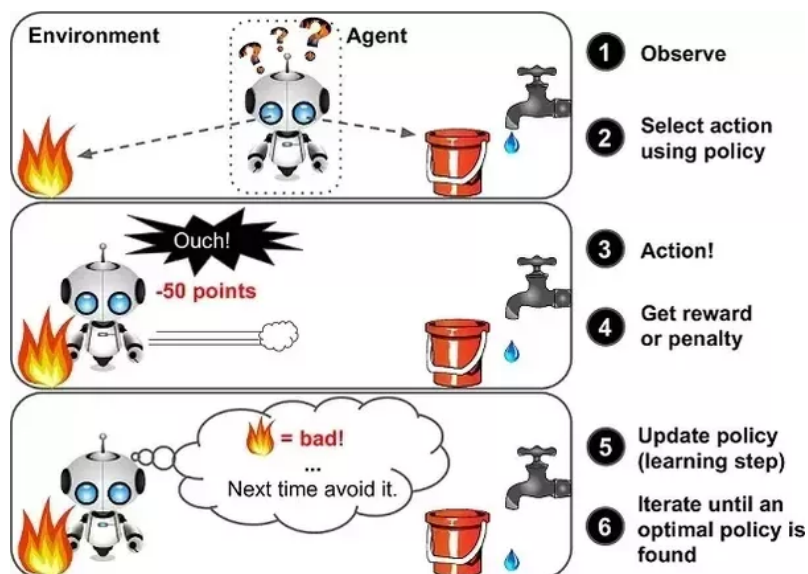


Abbildung 4.6: Agentenbasiertes Lernen [11]

Da es bei der Vorhersage von Retirement Curves keine Aufgabe zu bewältigen gibt, die ein Agent selbstlernend erfüllen kann und klassifizierte Trainingsdaten vorliegen, wird für die Vorhersage der Retirement Curves überwachtes Lernen verwendet.

4.3 Neuronale Netze

Neuronale Netze sind Vernetzungen aus künstlichen Neuronen, die den natürlichen Neuronen des menschlichen Körpers nachempfunden sind; keine genaue Nachbildung von Natürlichen Neuronalen Netzen (NNN), sondern eine Abstraktion zur Informationsverarbeitung.

Dieser von der Natur inspirierte Mechanismus ermöglicht es Computern ähnlich wie ein Gehirn zu lernen [25]. In den folgenden Abschnitten wird die Herkunft von Neuronalen Netzen erläutert, es wird die Funktionsweise beschrieben und das Training von Neuronalen Netzen wird vorgestellt.

4.3.1 Herkunft der Idee der Neuronalen Netze

Seit vielen Jahren strebt die Menschheit nach der Entwicklung einer intelligenten Maschine. Dabei geben wir uns nicht mit einfachen Aufgaben, wie Lasten heben oder Rechenaufgaben rechnen, zufrieden. Stattdessen sollen immer komplexere Aufgaben, wie das Klassifizieren von Bildern, die Vorhersage von Hauspreisen oder das Entwickeln eines Schachcomputers maschinell, gelöst werden. [37] Dieser Wunsch war so stark, dass schon im 18. Jahrhundert Geräte wie der „Schachtürke“ (Abbildung 4.7) entstanden, die eine intelligente Maschine vorgaukelten.

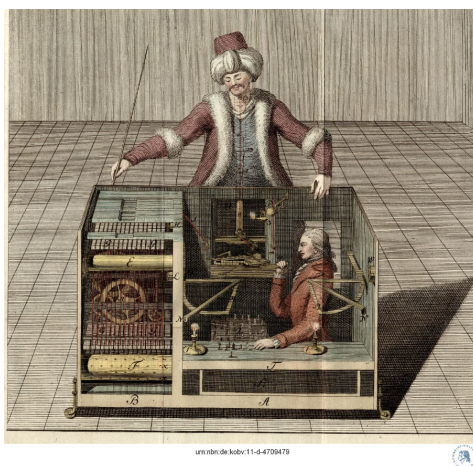


Abbildung 4.7: Der Schachtürke [48]

Ab der Mitte des 20. Jahrhunderts wurden die Computerbestandteile immer kostengünstiger. Dieser Trend zeichnet sich bis heute ab. Abbildung 4.8 zeigt die Preise von Arbeitsspeicher zwischen 1997 und 2011. Durch diese Preissenkung des Faktors 550 in nur 15 Jahren konnten immer leistungsfähigere und bezahlbarere Computer gebaut werden.

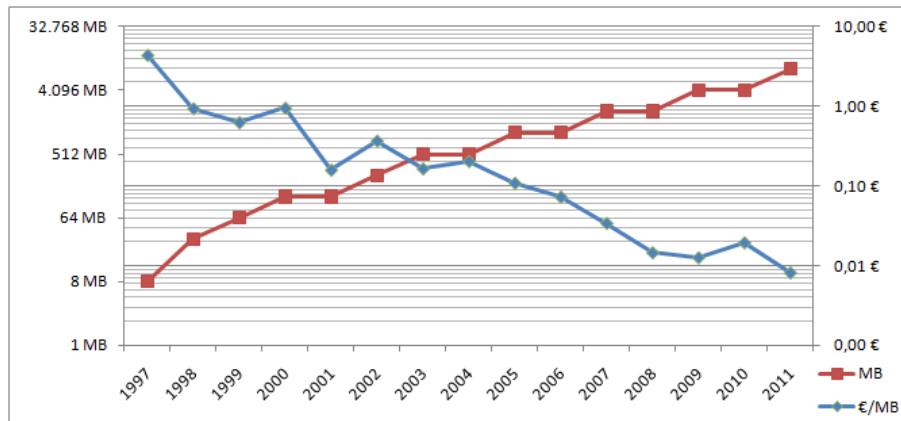


Abbildung 4.8: Arbeitsspeicher Preise zwischen 1997 und 2011 [4]

Damit kam die Idee auf, mit einem Computer ein Gehirn nachzubauen. [37] Diese, von der Natur inspirierten, Neuronalen Netze werden in den folgenden Abschnitten genauer dargestellt.

4.3.2 Aufbau eines Neurons

In diesem Abschnitt werden Natürliche Neuronen und die daraus abgeleiteten Künstlichen Neuronen vorgestellt.

Natürliches Neuron

Ein Natürliches Neuron ist eine Nervenzelle und somit ein Teil des Nervensystems des menschlichen oder tierischen Körpers.

Abbildung 4.9 zeigt den Aufbau eines Neurons. Es ist zu erkennen, dass die Nervenzelle viele kleine Auswüchse, sogenannte *Dendriten*, in der Nähe des Zellkerns hat. Darüber

empfängt das Neuron die Eingangssignale. Im Zellkern werden die Eingangssignale dann verarbeitet und ab einem bestimmten Schwellenwert wird ein Ausgangssignal erzeugt. Anders ausgedrückt: Das Neuron *feuert*. Dieses Ausgangssignal wird über das *Axon* an die *Terminale* weitergeleitet. Die *Terminale* können mit weiteren Neuronen verknüpft sein um die Informationen weiterzuleiten. [53]

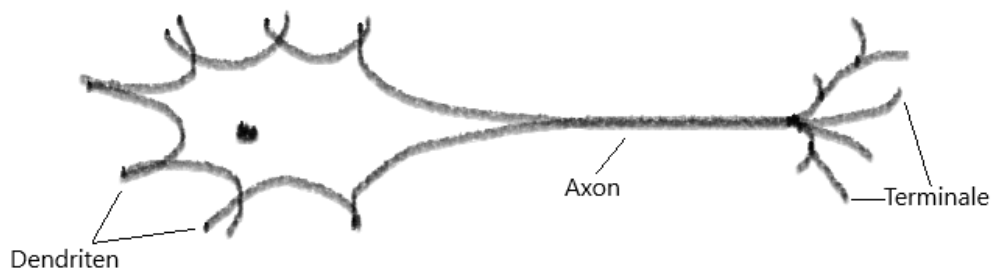


Abbildung 4.9: Aufbau eines Neurons

Künstliches Neuron

Ein Künstliches Neuron produziert aus den eingehenden Signalen ein Ausgangssignal. Das funktioniert, indem die Eingangssignale zunächst mit Gewichten gewertet und anschließend addiert werden. In Abhängigkeit des eingehenden Signals wird ein Ausgangswert erzeugt. Dieser wird mit einer sogenannten Aktivierungsfunktion ermittelt.

Abbildung 4.10 zeigt zwei Aktivierungsfunktionen im Vergleich. Die Sigmoid-Funktion verläuft zwischen 0 und 1 und ist eine logistische Funktion. Das bedeutet, dass die Funktion begrenzt ist und sich immer weiter dem Maximalwert annähert.

Die ReLU-Funktion (Rectified Linear Unit) hat im negativen Bereich immer den Wert von 0 und im positiven den Wert von x .

Beide Funktionen sind mögliche Aktivierungsfunktionen für ein Künstliches Neuron, da sie ab einem bestimmten Punkt ein gewisses Level übersteigen und somit das Neuron aktiviert wird und *feuert*. Welche der Funktionen verwendet wird, hängt vom jeweiligen Anwendungsfall ab. Ein Paper von Rasamoelina, Adjailia und Sincak [34]

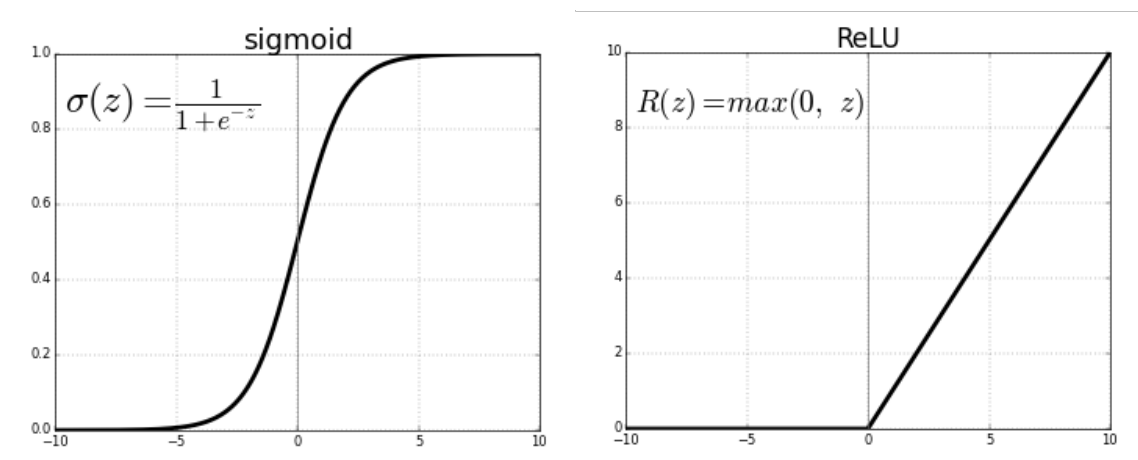


Abbildung 4.10: Sigmoid und ReLU Aktivierungsfunktion [1]

zeigt und vergleicht unterschiedliche Aktivierungsfunktionen für Neuronale Netze, die in den versteckten Schichten eines Neuronalen Netzes Verwendung finden.

4.3.3 Aufbau eines Neuronalen Netzes

Wie schon bei den Neuronen selber, liegt der Ursprung des Künstlichen Neuronalen Netzes (KNN) in der Natur. Daher wird in den folgenden Abschnitten zuerst die Struktur eines Natürlichen Neuronalen Netzes vorgestellt und daraufhin auf die Struktur des KNN eingegangen.

Natürliches Neuronales Netz

Als Natürliches Neuronales Netz bezeichnet man einen Verbund von Natürlichen Neuronen.

Abbildung 4.11 zeigt die Zeichnung eines Vogelgehirns. Es ist zu erkennen, dass die Neuronen miteinander verknüpft sind. Über diese Wege können Signale weitergeleitet werden. Erkennt eine Nervenzelle einen Reiz, wird ein Signal erzeugt und letztlich bis zum Gehirn weitergeleitet, wo es verarbeitet wird und eine finale Ausgabe erzeugt wird.

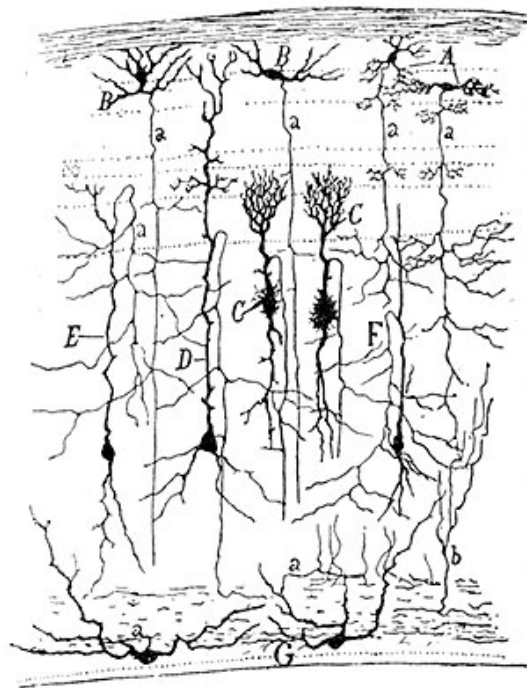


Abbildung 4.11: Zeichnung des Mittelhirns eines Spatzen [39]

Die Nerven des Vogels, die Reize wahrnehmen, sind **Input Layer** für dieses Neuronale Netz. Diese Reize werden durch die Neuronen (**Hidden Layers**) weitergeleitet. Im **Output Layer** entsteht ein elektrisches Signal, das zu einer Muskelbewegung (Flügelschlag, Kopf drehen, Hüpfen) führt.

Diese mehrschichtige Netzwerkarchitektur wird durch die Künstlichen Neuronale Netze nachgebaut, um basierend auf unterschiedlichen Eingabedaten im **Input Layer** eine entsprechende Ausgabe im **Output Layer** zu erzeugen.

Künstliches Neuronales Netz

Um die Mehrschichtigkeit eines Gehirns zu simulieren, sind Neuronale Netze aus verschiedenen Schichten (im Folgenden als **Layer** bezeichnet) zusammengesetzt. Die Daten, die vorhergesagt werden sollen oder die zum Training der Gewichte dienen, werden in der Eingangsschicht (**Input Layer**) in das Neuronale Netz eingeführt.

In der Ausgabeschicht (**Output Layer**) werden die berechneten Klassenzugehörigkeiten oder Regressionswerte ausgegeben.

Bei der Vorhersage von Klassenzugehörigkeiten wird zwischen der Wahrscheinlichkeit der Zugehörigkeit zu einer Klasse und der Vorhersage einer exakt zugeordneten Klasse unterschieden. Eine Regression liefert einen angenäherten Wert an den unbekanntem, tatsächlichen Wert. Damit können stetige Variablen, wie zum Beispiel die exakte Lebensdauer von Flugzeugen vorhergesagt werden.

Zwischen **Input Layer** und **Output Layer** befinden sich beliebig viele versteckte Schichten (**Hidden Layer**). Diese werden so genannt, da sie von außen nicht sichtbar sind. Sie empfangen keine Eingabe von außerhalb des Netzes und geben keine Ausgabe nach außen ab, tragen aber dennoch maßgeblich zur Berechnung bei.

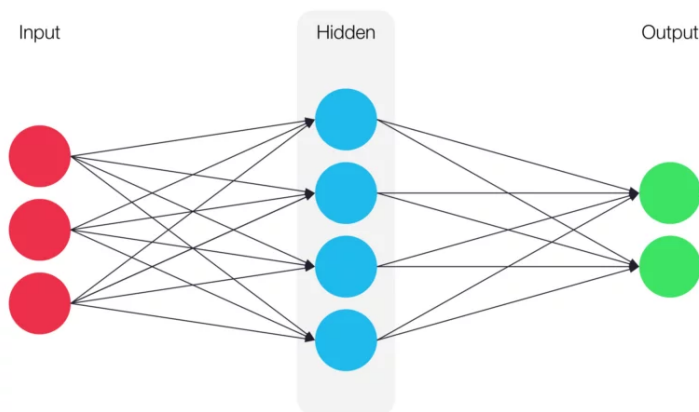


Abbildung 4.12: Struktur eines KNN [51]

Abbildung 4.12 zeigt den Aufbau eines KNN. Zu erkennen ist, dass jedes Neuron eine Verbindung zu jedem Neuron der Vor- sowie der Folgeschicht hat. Diese Struktur spiegelt die Verbundenheit eines Gehirns in der Natur wider. Der Output jedes Neurons wird also von allen Neuronen der Folgeschicht verwendet. Dabei wird der Output mit einem bestimmten Gewicht multipliziert, um so den Einfluss des Wertes zu steuern.

All diese Gewichte müssen fein justiert werden, damit der Einfluss jedes Eingangswertes angepasst wird, um die optimalen Vorhersagen berechnen zu können. Dies geschieht im Training des Neuronales Netzes mithilfe der sogenannten Backpropagation.

4.3.4 Backpropagation

Backpropagation ist ein Algorithmus, um den komplexen Gradienten der Kostenfunktion zu berechnen, welcher der Korrektur der Gewichte dient. Die genaue Funktionsweise der Backpropagation [14] wird hier nicht dargestellt, es wird nur eine kurze Zusammenfassung gegeben.

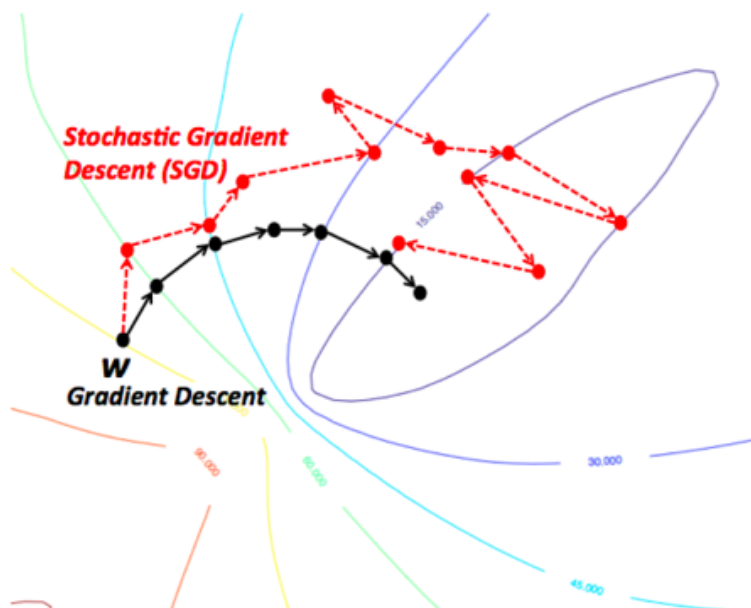


Abbildung 4.13: Stochastic Gradient Descent vs Gradient Descent [41]

Backpropagation ist der Algorithmus, um zu berechnen, wie ein einzelnes Trainingsdatum die Gewichte und Bias verändern will, um die eigene Vorhersage genauer zu machen. Das wird für alle Trainingsdaten durchgeführt und gemittelt, um die durchschnittliche gewünschte Änderung der Gewichte zu bestimmen.

Da bei großen Trainingsdatensätzen und vielschichtigen Neuronalen Netzen die Berechnung aller Gewichtsänderungen sehr langsam ist, wird oft der Stochastische Gradientenabstieg (SGD) dem herkömmlichen Gradientenabstieg vorgezogen.

Abbildung 4.13 zeigt den Stochastischen Gradientenabstieg im Vergleich zum herkömmlichen Gradientenabstieg. Es ist zu erkennen, dass letzterer zielstrebig und

direkter zum Ziel findet, wohingegen der SGD eher willkürliche Richtungen einnimmt und sich dem Zielwert über Umwege annähert. Dadurch, dass der SGD aber schneller berechnet werden kann, ist diese Methode im Allgemeinen weniger zeitaufwändig. Beim SGD wird der Datensatz in viele kleine Mini Batches eingeteilt und das Netz wird in kleineren Etappen trainiert. Da die Gewichte nicht mit allen Trainingsdaten gleichzeitig angepasst werden, ist der stochastische Gradientenabstieg weitaus schneller, jedoch ist die Genauigkeit dafür auch geringer. [29]

Im Gegensatz zum Training mit allen Daten, werden die Gewichte nicht immer in die exakt richtige Richtung angepasst, jedoch führt der stochastische Teil des Verfahrens zu einem guten Ergebnis.

4.4 TensorFlow und Keras

TensorFlow und Keras sind Open-Source Python Bibliotheken, die den Umgang mit Machine Learning und Neuronalen Netzen erleichtern.

TensorFlow

TensorFlow ist eine Open-Source-Plattform für Machine Learning, die vom Google Brain Team entwickelt wurde. Die Bibliothek wird unter der `Apache 2.0 open-source license` verteilt. [45]

TensorFlow bietet verschiedene Abstraktionslevel. Es können Modelle mithilfe der high-level Keras API erstellt werden, sodass sich nicht mit den Algorithmen im Detail befassen muss. Alternativ kann für mehr Flexibilität die Distribution Strategy API verwendet werden, um verteiltes Training zu ermöglichen. [46]

TensorFlow bietet Flexibilität und kann mittels einer Vielzahl von Algorithmen wie dem Training und der Inferenzbestimmung von Tiefen Neuronalen Netzen Tools für Machine Learning Systeme bereitstellen. Anwendungsgebiete sind unter anderem Spracherkennung, Computervision, Robotik, Natural Language Processing und Informationsbeschaffung. [2]

Keras

Keras ist eine Schnittstelle für Programmierer. Durch die Verwendung von konsistenten und einfachen Schnittstellen, sowie minimierten Benutzeraktionen, wird eine benutzerfreundliche Umgebung geschaffen und einer Überforderung durch Datenüberfrachtung entgegengewirkt. Über klare und eindeutige Fehlermeldungen wird den Entwicklern geholfen. [23] Des Weiteren verfügt es über eine ausführliche Dokumentation [21] und Developer Guides [22].

Mithilfe von Keras ist es sehr leicht möglich, Neuronale Netze zu erstellen und zu trainieren.

Listing 4.1: Zweischichtiges Neuronales Netz mit Keras

```
1 # Sequential API wird verwendet, um shape des Netzes
   festzulegen
2 model = keras.models.Sequential([
3     keras.layers.Dense(100, activation="relu", input_shape=X_train.
       shape[1:]), # Input Layer
4     keras.layers.Dense(2, activation="softmax") # Output Layer
5 ])
6 # Modell wird mit loss-Funktion und Optimisierer kompiliert
7 model.compile(loss="sparse_categorical_crossentropy", optimizer
   ="sgd")
8 # Modell wird mit vorher definierten Trainingsdaten in 20
   Epochen Trainiert
9 history = model.fit(X_train, y_train, epochs=20,
   validation_data=(X_valid, y_valid))
```

Listing 4.1 zeigt, wie ein zweischichtiges Neuronales Netz zur Klassifizierung von zwei Klassen erstellt wird. Es besteht aus zwei **Dense Layers**, die vollständig miteinander verknüpft sind. Die Eingangsdaten des Netzes haben die Form der Trainingsdaten. Die erste Schicht nach dem Input hat 100 Neuronen, das Output Layer hat nur zwei Neuronen, da genau zwei Klassen vorhergesagt werden sollen.

Mittels des Aufrufs `model.compile` (Zeile 7) wird das erzeugte Modell mit der `sparse_categorical_crossentropy` Loss-Funktion und dem Stochastischen Gradientenabstieg als Optimierer kompiliert.

In 20 Epochen werden dann die Gewichte und Bias des Netzes mit den Trainingsdaten trainiert (Zeile 9). Dabei werden Validierungsdaten übergeben, die jedoch nur dazu dienen, die Hyperparameter des Netzes unabhängig von den Trainingsdaten einzustellen und so eine Überanpassung zu verhindern.

Durch diese hohe Abstraktion bietet Keras nahezu unbegrenzte Möglichkeiten ein Neuronales Netz zu gestalten und eine Feinanpassung der Trainingsparameter vorzunehmen. Somit ist Keras eine geeignete Schnittstelle für TensorFlow um diverse Problemstellungen anzugehen.

5 Datenauswahl

In diesem Kapitel wird die Art der Vorhersage sowie die Auswahl der Trainingsdaten für das NN erläutert. Darüber hinaus wird auf Herausforderungen bei der Vorbereitung der Trainingsdaten eingegangen.

5.1 Art der Vorhersage

Die gewünschte Vorhersage ist die Wahrscheinlichkeit des retirements eines Flugzeuges in Abhängigkeit von Alter und anderen Faktoren, die in Abschnitt 5.2 genauer dargestellt werden.

Die Wahrscheinlichkeit ist ein Wert zwischen 0 und 1. Die Trainingsdaten verfügen nur über Zugehörigkeiten zu den jeweiligen Randwerten. Um eine Retirementwahrscheinlichkeit zu erhalten, wird die Wahrscheinlichkeit der Zugehörigkeit zu den beiden Klassen 0 und 1 vorhergesagt.

Somit wird wie in Abschnitt 4.3.3 beschrieben eine Regression durchgeführt, da ein stetiger Wert für die Wahrscheinlichkeit des retirements vorhergesagt wird.

Diese vorhergesagten Werte bilden, wenn sie in Abhängigkeit des Alters der Flugzeuge aufgetragen werden, eine Retirement Curve.

5.2 Trainingsdaten

Um das Neuronale Netz zu trainieren, wird die globale Flotte der Passagierflugzeuge zwischen 1955 und 2017 verwendet. Der Datensatz umfasst rund 50.000 Passagierflugzeuge mit Informationen über Flugzeugtyp, Baujahr, letztes Betriebsjahr und weitere Daten. Aus dem Datensatz der Flotte werden die folgenden Daten ausgewählt:

- Alter
- Flugzeugtyp

Die so extrahierten Daten werden mit den Wirtschaftsdaten des Baujahres angereichert, um eine Trainingsgrundlage für das Neuronale Netz zu schaffen. Bei den Wirtschaftsdaten handelt es sich um die folgenden Daten:

- Bruttoinlandsprodukt pro Kopf, global (GDP, *engl. Gross Domestic Product*)
- Ölpreis
- Passagierzahlen

Im Folgenden steht „Bruttoinlandsprodukt“ für „Bruttoinlandsprodukt pro Kopf“ und wird gleichbedeutend verwendet.

In den folgenden Absätzen werden die abgeschätzten Auswirkungen sowie die ermittelte Signifikanz der Eingabeparameter dargestellt.

5.2.1 Vermutete Auswirkungen der Eingabeparameter

Die vermuteten Auswirkungen der Eingabemerkmale werden in diesem Abschnitt aufgeschlüsselt. Die Auswirkungen sind in Tabelle 5.1 dargestellt und werden in den folgenden Absätzen erläutert.

5.2 Trainingsdaten

Eingabemerkmale	Veränderung Merkmal	Veränderung Überlebenswahrscheinlichkeit
Alter	↑	↓
GDP	↑	↑
Ölpreis	↑	↓
Passagierzahlen	↑	↑

Tabelle 5.1: Vermutete Auswirkungen der verwendeten Merkmale

Alter Die Retirementwahrscheinlichkeit in Abhängigkeit vom Alter ist leicht zu erklären. Je älter ein Flugzeug wird, desto eher wird es ausgemustert. Ein neues Flugzeug wird mit nahezu verschwindender Wahrscheinlichkeit ausgetauscht, es sei denn, es ist defekt oder hat einen Unfall.

Bei älteren Modellen steigt die Wahrscheinlichkeit an, da diese durch neuere, modernere Modelle ersetzt werden.

Flugzeugtyp Bei der Analyse der Trainingsdaten fällt auf, dass die Flugzeuge unterschiedlichen Typen angehören. Bei den Flugzeugklassen handelt es sich um folgende:

- Business Jet
- Helicopter
- Narrowbody
- Piston
- Regional Jet
- Supersonic
- Turboprop
- Widebody

5.2 Trainingsdaten

Werden die Daten getrennt voneinander visualisiert, ergeben sich unterschiedliche Graphen für die verschiedenen Flugzeugklassen.

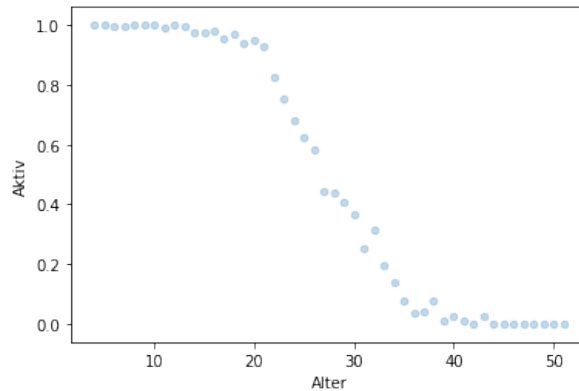


Abbildung 5.1: Trainingsdaten visualisiert (Widebody)

Abbildungen 5.1 und 5.2 zeigen die Trainingsdaten für Widebody und Turboprop Flugzeuge in visualisierter Form.

Dabei wird der Datensatz nach dem Alter der Flugzeuge gruppiert. Auf der y-Achse wird der Mittelwert der Spalte `Aktiv` aller Flugzeuge mit dem gleichen Baujahr aufgetragen. In der Spalte `Aktiv` wird mit einem Wahrheitswert (0 oder 1) festgehalten, ob ein Flugzeug noch aktiv ist.

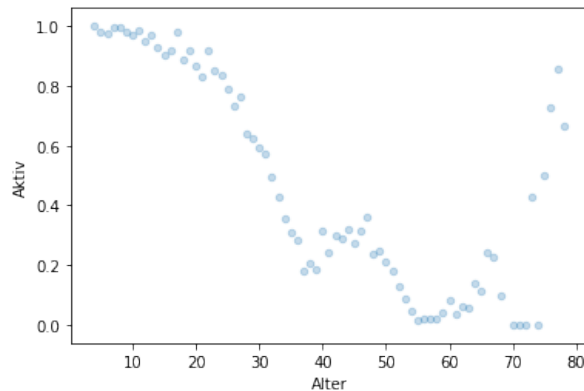


Abbildung 5.2: Trainingsdaten visualisiert (Turboprop)

Die Anomalien am hinteren Ende der Turboprop-Daten stammen daher, dass bei den Aufzeichnungen der ganz alten Flugzeuge nur noch die die verwendet werden erfasst sind. Für das Training werden diese Flugzeuge aus den Daten entfernt.

GDP Die Entwicklung des Bruttoinlandsprodukts beeinflusst die Lebensdauer der Flugzeuge. Da durch eine stärkere Wirtschaftskraft die Nachfrage nach Flügen steigt, müssen die Fluggesellschaften mehr Flugzeuge zur Verfügung stellen. Dadurch verschiebt sich die Entsorgung alter Flugzeuge.

Abbildung 5.3 zeigt die Entwicklung des weltweiten GDP in ganzen Zahlen und relativ zum Vorjahr. Weltweites GDP bedeutet, dass der Durchschnitt aller nationalen GDP eines Jahres bestimmt wird.

Dieser wird anhand von Daten der Worldbank [47] ermittelt. Dabei sind Phasen wirtschaftlichen Aufschwungs sowie Phasen mit geringerem Wachstum zu erkennen.

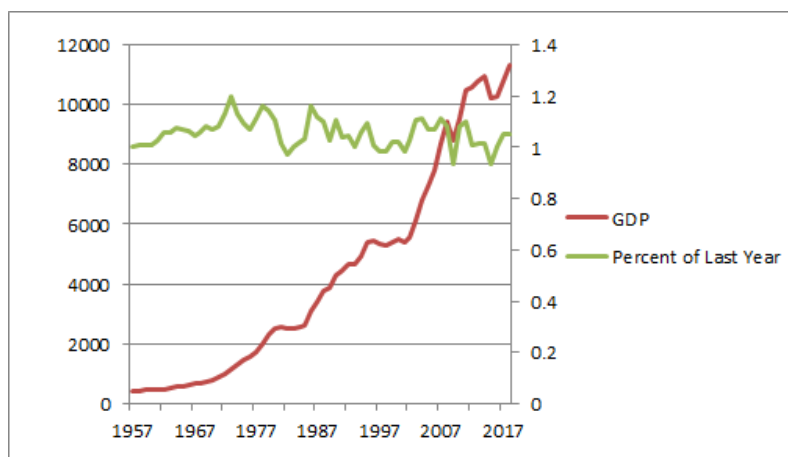


Abbildung 5.3: Entwicklung des GDP

Ölpreis Der Zusammenhang zwischen Ölpreis und der Lebensdauer von Flugzeugen ist antiproportional. Wenn der Ölpreis steigt, dann ist es wahrscheinlich, dass die Überlebensrate der Flugzeuge sinkt, da alte Flugzeuge einen höheren Treibstoffverbrauch haben, als neue Modelle.

Es liegt also im Interesse der Fluggesellschaften, die alten, verbrauchsintensiven Flugzeuge in ihrer Flotte durch neue, sparsamere Modelle zu ersetzen.

Die Entwicklung des Ölpreises in den letzten Jahren wird aus einem Aktienportal [40] übernommen und mit einer Datenerfassung von Tecson [44] abgeglichen.

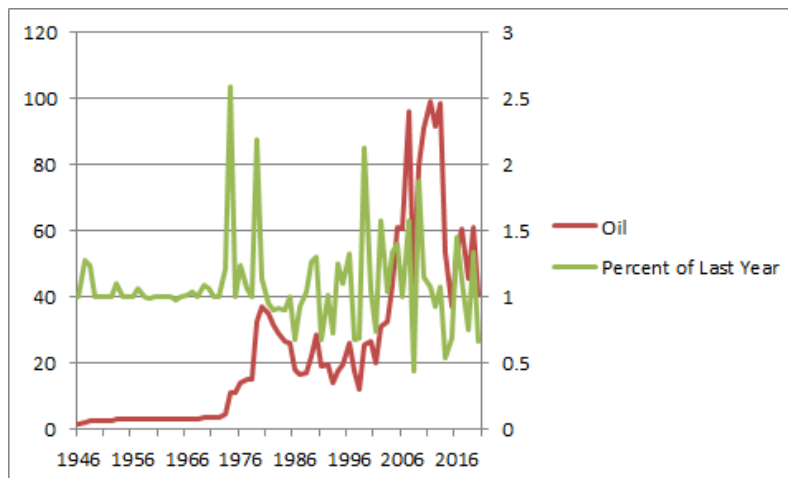


Abbildung 5.4: Entwicklung des Ölpreises

Wie in Abbildung 5.4 zu erkennen, ist der Ölpreis viel anfälliger für Schwankungen als das GDP. Der Ölpreis wird immer wieder durch Wirtschaftskrisen stark beeinflusst, aber auch technologische Fortschritte wie das umstrittene Fracking [19] können den Ölpreis drastisch absenken (in der Abbildung nach 2010).

Passagierzahlen Die Entwicklung der Passagierzahlen steht in Korrelation mit der Entwicklung des weltweiten Bruttoinlandsprodukts.

Steigende Passagierzahlen führen zu einer erhöhten Flugzeugnachfrage, sodass die Ausmusterungskriterien der Flugzeuge strenger werden und daraus resultierend das Retirement zurückgeht.

In Abbildung 5.5 werden die Passagierzahlen von 1950 bis 2018 dargestellt. Wie auch das Bruttoinlandsprodukt, steigen die Passagierzahlen seit dem Start der Aufzeichnung kontinuierlich an. Das liegt daran, dass Flugreisen für größere Teile der Bevölkerung wirtschaftlich möglich sind und mehr Flüge angeboten werden, indem neue Flughäfen gebaut und neue Routen geplant werden.

Die verwendeten Daten wurden von FW erfasst und aufbereitet.

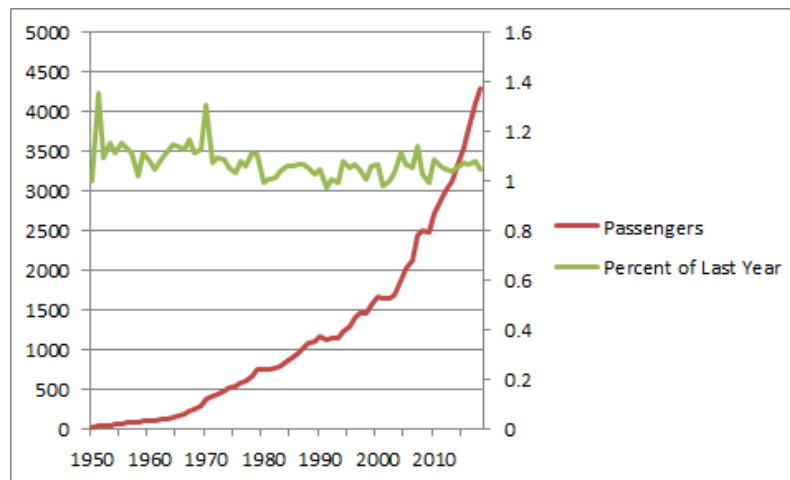


Abbildung 5.5: Entwicklung der Passagierzahlen

5.2.2 Signifikanz der Eingabeparameter

Für die Überprüfung der Signifikanz der Eingabeparameter werden verschiedene Neuronale Netze mit unterschiedlichen Parametern trainiert und verglichen. Dabei werden die in Tabelle 5.2 dargestellten Parameterkombinationen verwendet.

Nummer	Kombination	erwartete Performance
1	GDP, Oil Price, Passengers	sehr gut
2	Oil Price, Passengers	etwas schlechter
3	GDP, Passengers	etwas schlechter
4	GDP, Oil Price	etwas schlechter
5	GDP	deutlich schlechter
6	Oil Price	deutlich schlechter
7	Passengers	deutlich schlechter
8	-	schlecht
9	nur Jahre 0 - 20 nach Baujahr	deutlich schlechter
10	nur Jahre 10 - 30 nach Baujahr	etwas schlechter

Tabelle 5.2: Parameterkombinationen für die Signifikanzanalyse

Beim Vergleich der dargestellten Kombinationen stellt sich heraus, dass alle trainierten Netze sehr gute Vorhersagen treffen. Für den Vergleich wird eine Funktion

entwickelt, die für eine Vorhersage die Abweichung zum erwarteten Wert berechnet. Für jede der in Tabelle 5.2 beschriebenen Kombinationen wird die Abweichung zum erwarteten Wert berechnet und die Ergebnisse werden miteinander verglichen. Dabei stellt sich heraus, dass es die Performance weniger als angenommen beeinflusst, wenn die Anzahl der Eingabeparameter nach und nach reduziert wird.

Bei den Tests stellte sich heraus, dass das Hinzufügen beziehungsweise Entfernen von Eingabeparametern keinen positiven Einfluss auf die Qualität der Vorhersage hat. Daher werden für die Vorhersage der Retirement Curves alle vorhandenen Wirtschaftsdaten verwendet.

5.3 Vorbereitung der Trainingsdaten

Bei der Vorbereitung der Trainingsdaten wird zwischen Textmerkmalen und Zahlenmerkmalen unterschieden.

Zahlenmerkmale können von Computern direkt verarbeitet werden, wohingegen Textmerkmale erst in ein für den Computer lesbares Format übersetzt werden müssen. Dafür kann die sogenannte **One-Hot-Codierung** verwendet werden. Bei der **One-Hot-Codierung** wird für jeden Wert, den die zu codierende Spalte annehmen kann, eine neue Spalte erzeugt. In dieser wird festgehalten, ob der Wert vor der Codierung im Datensatz enthalten war. Dies wird mit den Ziffern 0 oder 1 gekennzeichnet.

Zahlenmerkmale können normiert (auf Intervall von 0 bis 1) oder standardisiert werden (Mittelwert von 0 mit Standardabweichung von 1), um die Bedeutung von Ausreißern zu minimieren und ungewöhnliche Werte zu bestrafen.

Als Vorbereitung für das Training des Neuronalen Netzes wird die Spalte mit dem Flugzeugtyp per **One-Hot-Codierung** codiert. Die restlichen Daten werden mit einem Standard-Scaler standardisiert, damit Ausreißer und Anomalien in den Trainingsdaten nicht so stark ins Gewicht fallen und die Prognose nicht verfälschen.

5.4 Schwierigkeiten bei der Datenauswahl

Bei der Datenauswahl sind zwei größere Herausforderungen aufgetreten. Die erste Schwierigkeit ist die Einbindung der Wirtschaftsdaten in die Trainingsdaten, die andere Schwierigkeit ist die Entscheidung, ob feste Wirtschaftsdaten oder die Veränderungsdaten verwendet werden sollen.

5.4.1 Einbinden von Wirtschaftsdaten

Für die Lebensdauer eines Flugzeuges sind nicht die Wirtschaftsdaten im Baujahr, sondern die Wirtschaftsdaten des letzten Betriebsjahres entscheidend. In der Regel sind die entstehenden Kosten maßgeblich für die Entscheidung des Aussortierens eines Flugzeuges. Bei zu großer Kosteneffizienz (zum Beispiel durch zu hohen Treibstoffverbrauch) ist daher das letzte Betriebsjahr für das Retirement verantwortlich. Daraus resultiert das folgende Problem:

Für Flugzeuge in den Trainingsdaten, die noch in Verwendung sind, ist das letzte Betriebsjahr nicht bekannt. Gleiches gilt für die Flugzeuge, für welche die Überlebenswahrscheinlichkeit vorhergesagt werden soll. Daher kann das letzte Betriebsjahr nicht als Merkmal verwendet werden.

Um dieses Problem zu umgehen, werden anstelle der Wirtschaftsdaten im letzten Betriebsjahr die Wirtschaftsdaten im Baujahr und jeweils 10, 20 und 30 Jahre später verwendet, um eventuelle Abhängigkeiten zu erkennen.

Diese Werte liegen teilweise in der Zukunft und sind daher nicht bekannt. Mithilfe von mehreren Szenarien für die Entwicklung der Wirtschaftsdaten sollen die fehlenden Werte möglichst realistisch bestimmt werden. Die Entwicklung dieser Szenarien wird in Abschnitt 5.5 beschrieben.

5.4.2 Wahl zwischen festen Werten und Veränderungsdaten

Bei der Validierung der ersten Testergebnisse ergab sich die Frage, ob die tatsächlichen Zahlen oder die Wirtschaftsdaten für das Training des Neuronalen Netzes verwendet werden sollen.

Vorliegend werden die Veränderungsdaten der Wirtschaftsdaten verwendet, da an-

sonsten durch die Inflation eine Scheinkorrelation entstehen würde. Die Entscheidungsfindung wird in Abschnitt 8.3 genauer dargestellt.

5.5 Entwicklung von Szenarien

Für die Entwicklung des GDP, des Ölpreises und der Passagierzahlen werden jeweils zwei Vorhersagen verwendet. Eine Vorhersage, die starkes Wachstum prognostiziert und eine Vorhersage, die ein schwächeres Wachstum prognostiziert. Wenn alle möglichen Kombinationen aus den Vorhersagen gebildet werden, ergeben sich die folgenden acht Szenarien:

Szenario Nummer	Wachstum GDP	Wachstum Ölpreis	Wachstum Passagiere
1	Niedrig (2,1%)	Niedrig (2,5%)	Niedrig
2	Niedrig (2,1%)	Niedrig (2,5%)	Hoch
3	Niedrig (2,1%)	Hoch (3,5%)	Niedrig
4	Niedrig (2,1%)	Hoch (3,5%)	Hoch
5	Hoch (3,1%)	Niedrig (2,5%)	Niedrig
6	Hoch (3,1%)	Niedrig (2,5%)	Hoch
7	Hoch (3,1%)	Hoch (3,5%)	Niedrig
8	Hoch (3,1%)	Hoch (3,5%)	Hoch

Tabelle 5.3: Kombination der Szenarien

Die Prognose des weltweiten GDP orientiert sich an einer Untersuchung des CO2 Ausstoßes des weltweiten Flugverkehrs [30], sowie an der Dokumentation des Modells zur weltweiten Energieentwicklung der International Energy Agency (IEA) [18] und an einem Report über die kanadischen Energieressourcen [49].

Die Wachstumsrate des Ölpreises wird, ähnlich der des GDP, anhand der Dokumentation des Modells zur weltweiten Energieentwicklung [18] und dem Ausblick auf die Report zu den kanadischen Energieressourcen [49] modelliert.

Die Prognose der Passagierzahlen ist eine eigene Vorhersage von FW. Die Werte sind in Abständen von fünf Jahren vorhergesagt. Daher müssen die fehlenden Werte mithilfe einer linearen Interpolation (Gleichung (5.1)) ermittelt werden.

5.5 Entwicklung von Szenarien

$$x_{\text{jahr}} = x_{\text{startjahr}} + ((x_{\text{endjahr}} - x_{\text{startjahr}}) * \frac{\text{jahr} - \text{startjahr}}{\text{endjahr} - \text{startjahr}}) \quad (5.1)$$

Die Differenz der Passagierzahlen über eine Zeitspanne von fünf Jahren ($\text{endjahr} - \text{startjahr}$) wird zu einem Teil ($\frac{\text{jahr} - \text{startjahr}}{\text{endjahr} - \text{startjahr}}$) auf die Passagierzahlen des Startjahres aufaddiert, um die Werte für das gewünschte Jahr zu erhalten.

Aus diesen Zahlen lassen sich dann die Veränderungsraten der Passagierzahlen ermitteln.

6 Analyse und Konzeption

In diesem Kapitel werden die Anforderungen an das Programm analysiert und es wird eine Programmstruktur konzeptioniert. Außerdem werden die Inputs und Outputs, sowie die zu entwickelnden Artefakte spezifiziert.

6.1 Anforderungsanalyse

Es soll ein Programm entwickelt werden, das mithilfe eines Neuronalen Netzes Retirement Curves für Flugzeuge vorhersagt. Dabei werden die folgenden Anforderungen an das Programm gestellt:

- Die Daten sollen in Form einer CSV-Datei in das Programm eingegeben werden.
- Es dürfen nur Daten in einem vorgegebenen Format in das Programm eingegeben werden.
- Die Daten sollen automatisch in eine für das NN passende Form gebracht werden.
- Es soll ein Modell ausgewählt werden können, mit dem die Daten vorhergesagt werden.
- Das Programm soll als Ausgabe eine Matrix mit Werten erzeugen können.
- Das Programm soll Retirement Curves erstellen können.

6.2 Programmstruktur

Aufgrund der oben genannten Anforderungen wird eine Programmstruktur konzeptioniert.

6.2 Programmstruktur

Abbildung 6.1 zeigt die Struktur, nach der das Programm aufgebaut ist. Die Daten werden mit einer CSV-Datei in das Programm eingespeist. Daraufhin wird automatisch das richtige Modell für die Vorhersage ausgewählt und die Eingabedaten werden für die Prognose vorbereitet. Die vorhergesagten Daten können wahlweise in einer Matrix oder als Retirement Curve grafisch dargestellt werden. Die einzelnen Schritte werden als Python Funktionen realisiert, um mehrfach verwendbar und skalierbar zu sein.

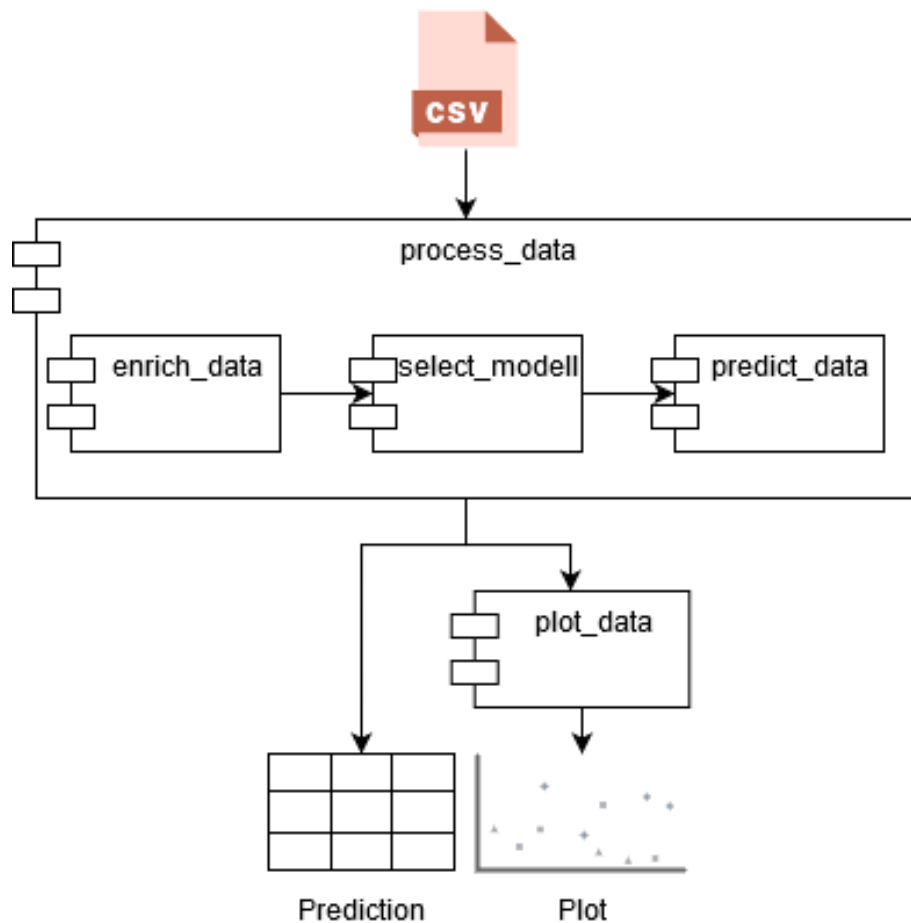


Abbildung 6.1: Programmstruktur

6.3 Spezifikationen

Das Ziel der Arbeit ist die Entwicklung eines Python Skripts, das aus Input-Daten, sofern sie in einer fest definierten Form vorliegen, eine Vorhersage prognostiziert und Output-Daten erzeugt. Die Output-Daten sollen entweder als Retirement Curve dargestellt werden oder eine Matrix mit den vorhergesagten Daten bilden.

Dafür werden vier Funktionen definiert:

- Eine Funktion dient der Vorbereitung der Eingangsdaten.
- Eine Funktion dient der Auswahl und dem Laden des Neuronalen Netzes, das mit dem gewünschten Modell trainiert wurde.
- Eine Funktion dient der Vorhersage der Daten mit dem ausgewählten Modell.
- Eine Funktion dient der grafischen Visualisierung der Vorhersageergebnisse.

Input-Daten

Die Input-Daten, die per CSV-Datei in das Programm eingegeben werden, sollen die folgende Form haben:

Aircraft Class	Construction Year	Reference Year	GDP	Oil Price	Passenger
Widebody	1970	2020	high	high	low
Widebody	1980	2020	high	high	low
Widebody	1990	2020	high	high	low

Tabelle 6.1: Format der Eingabedaten

Daten in dieser Form (Tabelle 6.1) werden mithilfe einer Funktion in eine für das Neuronale Netz verwendbare Form gebracht.

Output-Daten

Der Output des Neuronalen Netzes ist eine gewisse Wahrscheinlichkeit zwischen 0 und 1, mit der das beschriebene Flugzeug noch aktiv ist. Die Daten werden als Matrix ausgegeben. Dabei werden die Informationen über das Flugzeug wie in Tabelle 6.1 zu

6.3 Spezifikationen

sehen, dargestellt und es wird eine neue Spalte für die Überlebenswahrscheinlichkeit erstellt. Alternativ können die Daten nach Alter und Typ sortiert in einem Graphen aufgetragen werden. Dabei entsteht eine Retirement Curve. Abbildung 6.2 zeigt, wie die vom Neuronalen Netz vorhergesagten Retirement Curves aussehen sollen.

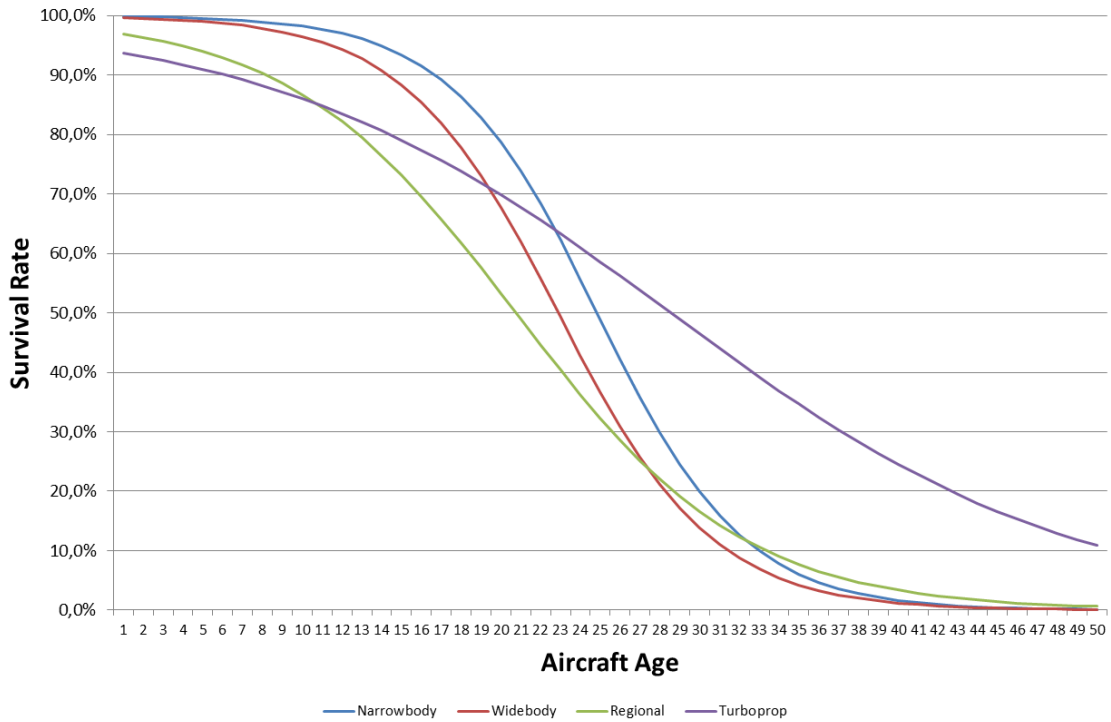


Abbildung 6.2: Gewünschte Ausgabe des Neuronales Netzes

7 Umsetzung

In diesem Kapitel wird das Training der Neuronalen Netze sowie die Implementierung des Programms für die Vorhersage der Retirement Curves beschrieben.

7.1 Training der Neuronalen Netze

Für die Umsetzung der definierten Anforderungen werden acht Neuronale Netze erstellt. Jedes der Neuronalen Netze wird mit einer anderen Datenbasis trainiert. Dadurch werden acht verschiedene Netze für die in Tabelle 5.3 beschriebenen Szenarien generiert.

Das Training der Netze läuft dabei nach den folgenden Schritten ab:

7.1.1 Trainingsdaten vorbereiten

Zunächst müssen die Trainingsdaten vorbereitet werden, damit das Neuronale Netz trainiert werden kann. Ein wichtiger Schritt dabei ist die Anreicherung des Datensatzes mit den Wirtschaftsdaten. Dabei wird aus einer für das Szenario vorbereiteten CSV-Datei ein Datensatz in das Python Programm eingelesen. Die Eingangsdaten haben dabei die in Tabelle 6.1 beschriebene Form. Damit die Daten vom Neuronalen Netz verwendet werden können, müssen aus der Beschreibung des Szenarios die tatsächlichen Werte ermittelt und eingesetzt werden. Dazu wird die folgende Umwandlung vorgenommen:

Build Year	Reference Year	GDP	→	Age	GDP0	GDP10	GDP20	GDP30
1990	2017	High		27	1.107	1.018	1.081	1.021
1990	2017	Low		27	1.107	1.018	1.081	1.031

Tabelle 7.1: Ersetzen von Entwicklungsmodellen durch reale Werte

Abschnitt 7.1.1 beschreibt die Ersetzung der Spalte **GDP** durch die Veränderungsrate des GDP im Baujahr, sowie in den folgenden 10, 20 und 30 Jahren.

Des Weiteren wird eine **One-Hot-Codierung** auf das Merkmal **Aircraft Class** angewandt. Dadurch wird für jeden möglichen Wert der Spalte **Aircraft Class** eine neue Spalte erzeugt, die den Wert 1 oder 0 annehmen kann. Hiermit sind die Eingabedaten für das Training der Neuronalen Netze vorbereitet. Die endgültige Anzahl der Spalten der Trainingsdaten beträgt 21:

- Aircraft Age (1)
- Aircraft Class (8)
- GDP (4)
- Oil Price (4)
- Passenger (4)

Eine weitere Spalte beinhaltet den vorherzusagenden Wert. Diese Daten müssen dem Neuronalen Netz für das Training übergeben werden.

7.1.2 Optimieren der Netze

Die trainierten Neuronalen Netze werden durch die geschickte Wahl von Hyperparametern (Optimierer, Anzahl der Epochen, Anzahl der Knoten) optimiert.

Im Programm werden mehrere Neuronale Netze verwendet. Ein Netz davon wird analysiert und optimiert, die anderen Netze werden mit den gleichen Hyperparametern trainiert, um den Aufwand möglichst gering zu halten.

Optimierung der Form des Netzes Es werden acht verschiedene Netzwerkformen miteinander verglichen. Dabei wird eine Kombination aus Schichten mit unterschiedlicher Anzahl an Neuronen und Layern verwendet. Außerdem werden Dropout Layer mit unterschiedlichen Dropoutraten genutzt. Die acht verwendeten Netzwerkformen

7.1 Training der Neuronalen Netze

werden in Tabelle 7.2 dargestellt.

Modell	Shape
0	Input Layer (10, relu), Output Layer (2, softmax)
1	Input Layer (100, relu), Dense Layer (100, relu), Output Layer (2, softmax)
2	Input Layer (100, relu), Dropout Layer (0.2), Dense Layer (100, relu), Output Layer (2, softmax)
3	Input Layer (100, relu), Dropout Layer (0.5), Dense Layer (100, relu), Output Layer (2, softmax)
4	Input Layer (100, relu), Dense Layer (50, relu), Dense Layer (10, relu), Output Layer (2, softmax)
5	Input Layer (200, relu), Dense Layer (100, relu), Output Layer (2, softmax)
6	Input Layer (100, relu), Dropout Layer (0.2), Dense Layer (50, relu), Dropout Layer (0.2), Dense Layer (10, relu), Output Layer (2, softmax)
7	Input Layer (500, relu), Dense Layer (250, relu), Output Layer (2, softmax)

Tabelle 7.2: Formen des Neuronalen Netzes

7.1 Training der Neuronalen Netze

Die verschiedenen Netzwerkformen werden kompiliert und mit den Trainingsdaten trainiert. Anschließend wird ein vorbereiteter Testdatensatz von jedem Netzwerk vorhergesagt und die Ergebnisse werden miteinander, sowie mit dem erwarteten Ergebnis verglichen.

Der Testdatensatz umfasst Daten zu Narrowbodyflugzeugen verschiedenen Alters (Tabelle 7.3). Wenn die Daten visualisiert werden, sollten sie eine Retirement Curve bilden.

Aircraft Class	Build Year	Reference Year	GDP	Oil Price	Passenger
Narrowbody	1970	2017	Low	Low	Low
Narrowbody	1975	2017	Low	Low	Low
Narrowbody	1980	2017	Low	Low	Low
Narrowbody	1985	2017	Low	Low	Low
Narrowbody	1990	2017	Low	Low	Low
Narrowbody	1995	2017	Low	Low	Low
Narrowbody	2000	2017	Low	Low	Low
Narrowbody	2005	2017	Low	Low	Low
Narrowbody	2010	2017	Low	Low	Low
Narrowbody	2015	2017	Low	Low	Low
Narrowbody	2017	2017	Low	Low	Low

Tabelle 7.3: Testdaten für die Optimierung

Nachdem die Modelle kompiliert und trainiert sind, werden die Testdaten vorhergesagt und in einem Graphen dargestellt. Abbildung 7.1 zeigt, dass alle Modelle bei den Narrowbody Flugzeugen ungefähr die gleiche Retirement Curve vorhersagen, ein Modell jedoch stark abweicht.

Die anderen Modelle konvergieren sehr gut. Das Modell 3 fällt jedoch negativ auf. Der Grund für die schlechte Performance ist möglicherweise der zu hohe Wert im Dropout Layer. Wenn 50% der Neuronen nicht mit der folgenden Schicht verknüpft sind, gehen zu viele Informationen verloren, als dass eine gute Vorhersage getroffen werden kann.

7.1 Training der Neuronalen Netze

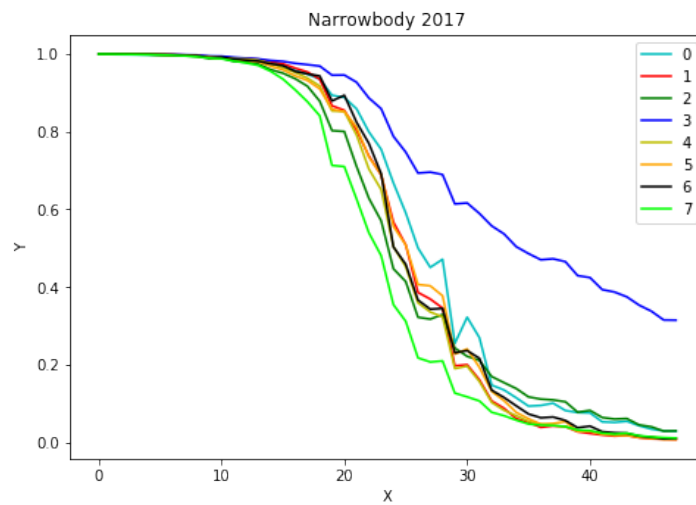


Abbildung 7.1: Vorhersage für Narrowbody Flugzeuge

Beim Vergleichen der Vorhersage für Widebody Flugzeuge (Abbildung 7.2), fällt auf, dass auch für diese (hier: Boeing 747) die Modelle sehr gut konvergieren. Jedoch ist auch hier Modell 3 deutlich schlechter als die anderen. In der Mitte (20-30 Jahre) ist der Abstand zwischen den Vorhersagen der Modelle jedoch etwas größer als bei Flugzeugen des Narrowbody Typs.

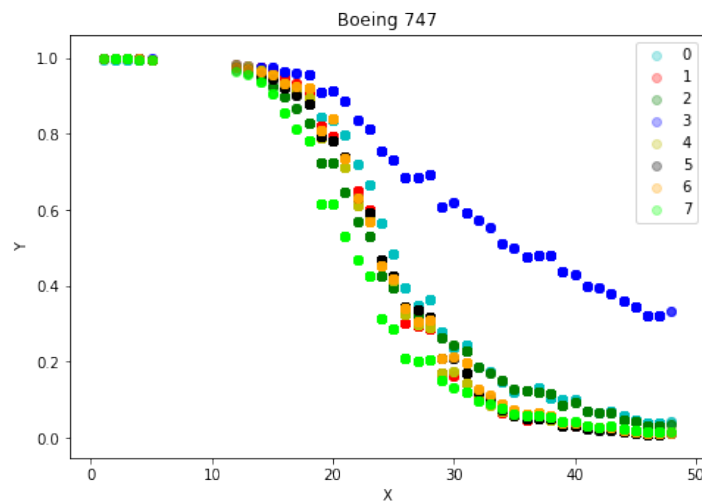


Abbildung 7.2: Vorhersage für Widebody Flugzeuge

Insgesamt gibt Modell 0 die optimistischste Schätzung ab, während Modell 7 die pessimistischste Vorhersage liefert.

Alles in allem liefern aber alle trainierten Modelle, mit Ausnahme des Modells 3, eine gute Vorhersage, die sich nur gering in ihrer Performance unterscheiden.

Optimierung der Kompilierungsparameter Beim Kompilieren der Modelle werden eine Kostenfunktion (engl. loss function) sowie ein Optimierer (engl. optimizer) gewählt. Ziel des Trainings ist es diese Kostenfunktion zu minimieren. Dafür werden die Gewichte mit dem `optimizer` angepasst.

Um das Neuronale Netz zu optimieren, werden die in Tabelle 7.4 aufgelisteten Parameter verwendet.

Loss function	Optimizer
Sparse Categorical Crossentropy	Stochastic Gradient Descent
Sparse Categorical Crossentropy	Adam
Categorical Crossentropy	Stochastic Gradient Descent
Categorical Crossentropy	Adam
Binary Crossentropy	Stochastic Gradient Descent
Binary Crossentropy	Adam

Tabelle 7.4: Kombination der Kompilierungsparameter

Auffällig bei den Testergebnissen ist, dass sich die `optimizer` untereinander kaum unterscheiden, die Kostenfunktionen jedoch einen großen Einfluss auf das Ergebnis haben.

Sowohl das Modell, das mit der `Categorical Crossentropy` trainiert wurde, als auch das Modell, das mit der `Binary Crossentropy` trainiert wurde, konvergieren nicht. Das neuronale Netz entscheidet sich immer für eine 50 prozentige Überlebenswahrscheinlichkeit.

Dies entspricht nicht der Realität. Somit sind die beiden Kostenfunktionen ungeeignet, um die Retirement Curves vorherzusagen. Festzuhalten ist allerdings, dass die beiden `optimizer` beide gut geeignet sind, um die Gewichte des Neuronalen Netzes feinzustimmen.

Optimierung der Trainingsepochen Die Anzahl der Trainingsepochen ist ausschlaggebend dafür, wie lange das Netz trainiert wird. Dabei gilt nicht immer, dass mehr Epochen zu einem besseren Ergebnis führen. Ein zu langes Training führt unweigerlich zu einer Überanpassung des Neuronalen Netzes an die Trainingsdaten. Das ist nicht gewünscht, da die Vorhersagen durch eine Überanpassung verfälscht werden.

Ein Durchlauf mit 10, 20 und 30 Trainingsepochen zeigt, dass die Ergebnisse sich noch nicht stark unterscheiden.

Bei den Trainingsepochen muss auch darauf geachtet werden, dass die Durchlaufzeit des Trainings des Neuronalen Netzes nicht zu stark beeinträchtigt wird.

Daher ist eine Anzahl von 20 Trainingsepochen ein guter Kompromiss zwischen Laufzeit und Vorhersagegenauigkeit, ohne dabei an die Trainingsdaten überangepasst zu sein.

7.1.3 Spezifikationen der Neuronalen Netze

In diesem Abschnitt wird die Form der Neuronalen Netze dargestellt und die Trainingsumgebung wird beschrieben.

Für die finale Prognose werden Neuronale Netze verwendet, welche die folgende Form haben:

- 21 input neurons
- 10 hidden neurons (Activation: ReLU)
- 2 output neurons (Activation: Softmax)

Die Netze werden auf einem Laptop mit 8 GB RAM und einem Intel(R) Core(TM) i5-6300HQ Prozessor trainiert. Da die Netze sehr klein sind und in 20 Epochen trainiert werden, dauert das Training eines Netzes ca. 40-50 Sekunden.

Während der Optimierung werden auch andere Netzwerkformen verwendet, sodass das Training bis zu einigen Minuten andauern kann.

7.1.4 Speichern der Netze

Das Training der Neuronale Netze nimmt trotz der geringen Größe einige Zeit in Anspruch und durch die zufällige Initialisierung der Gewichte zu Trainingsbeginn entsteht nicht immer das gleiche Neuronale Netz.

Daher sollen die Neuronale Netze nur einmal trainiert und dann abgespeichert werden, damit immer mit den gleichen NN vorhergesagt werden kann.

`Keras` implementiert eine Serialisierung für Neuronale Netze. Das Modell wird in einer YAML-Datei abgespeichert, das heißt die Informationen wie Format der Eingabedaten, Anzahl der Schichten und Anzahl der Knoten pro Schicht können mithilfe von `Keras` wieder geladen werden.

Die Gewichte, die trainiert wurden, um die Vorhersage des Neuronale Netzes zu optimieren, werden dabei allerdings nicht gespeichert. Jedoch können sie mithilfe von `Keras` in ein Hierarchical Data Format (HDF5) exportiert werden.

Ein weiteres Objekt, das gespeichert werden muss, ist der `StandardScaler`, mit dem die Eingabedaten skaliert werden. Damit im Betrieb des Programms die Vorhersageergebnisse nicht verfälscht werden, muss derselbe `StandardScaler` wie für die Trainingsdaten verwendet werden.

Mithilfe der Python-Bibliothek `pickle` [33] können Objekte serialisiert werden. Jeder `StandardScaler` der acht Neuronale Netze wird so gespeichert und kann für die Vorhersage wieder geladen werden.

Das Laden des Neuronale Netzes und des `StandardScalers` wird in Abschnitt 7.2 beschrieben.

7.2 Implementierung der Vorhersage

Für die Vorhersage mithilfe der Neuronale Netze werden vier Funktionen implementiert, die die einzelnen Schritte in der Verarbeitungskette darstellen. In den folgenden Abschnitten werden die Funktionen detailliert vorgestellt.

Einlesen der Daten

Der erste Schritt ist das Einlesen der Daten. Die Daten werden dem Programm in einer CSV-Datei zur Verfügung gestellt. Mithilfe der Python-Bibliothek `pandas` lassen sich CSV-Dateien als `DataFrame`-Objekt in ein Python Programm importieren. Die importierten Daten haben die in Abschnitt 6.3 beschriebene Form. Diese ist noch nicht für eine Vorhersage mit dem Neuronalen Netz geeignet, daher müssen die Daten vorbereitet werden.

Vorbereitung der Daten

Die Vorbereitung der Daten wird von der ersten implementierten Funktion übernommen. Die Funktion `enrich_data` wandelt Daten in ein für das Neuronale Netz lesbares Format um.

Basierend auf den gewählten Entwicklungsmodellen für Bruttoinlandsprodukt, Ölpreis und Passagierzahlen werden die entsprechenden Werte aus vorbereiteten Lookup-Tabellen entnommen und in den Datensatz eingesetzt.

Aus Baujahr und Referenzjahr wird das Alter des Flugzeugs ermittelt. Außerdem werden die Spalten für die One-Hot-Codierung des Flugzeugtyps erzeugt.

Die so vorbereiteten Daten haben damit die benötigten 21 Spalten wie in Abschnitt 7.1 beschrieben.

Modellauswahl

Anhand der gewählten Entwicklungsmodelle wird in der Funktion `load_scaler_and_model` das geeignete Neuronale Netz ausgewählt. Das gewünschte Modell wird in einer zusätzlichen Spalte an die Trainingsdaten angehängt und vor der Vorhersage beziehungsweise dem Training ausgelesen und entfernt. Mithilfe der von `Keras` implementierten Methoden werden das Modell und die Gewichte geladen. Das Modell muss mit den Gewichten kompiliert werden, bevor es für die Vorhersage verwendet werden kann. Das kompilierte Modell und der geladene `StandardScaler` werden von der Funktion zurückgegeben, sodass sie im Programmablauf von der Funktion `process_dataframe` für die Vorhersage verwendet werden können.

Vorhersage

Der Funktion `process_dataframe` für die Vorhersage der Daten wird als Parameter ein `DataFrame`-Objekt übergeben.

Innerhalb der Funktion werden die darin übergebenen Daten mithilfe der Funktion `enrich_data` in die richtige Form gebracht. Daraufhin werden mit der Funktion `load_scaler_and_model` das trainierte Modell und der `StandardScaler` für die Daten geladen. Die vorherzusagenden Daten werden mit dem `StandardScaler` skaliert und dann dem trainierten Modell übergeben.

Das Neuronale Netz ermittelt anschließend für jeden Datensatz im `DataFrame` eine Überlebenswahrscheinlichkeit und gibt eine Liste mit allen vorhergesagten Überlebenswahrscheinlichkeiten zurück.

Ausgabe

Die von der Funktion `process_dataframe` ermittelten Daten können entweder direkt als Matrix verwendet werden oder mit der Funktion `plot_retirement_curve` als Retirement Curve grafisch dargestellt werden.

Dafür wird mit der Python-Bibliothek `matplotlib` ein Graph erzeugt. Auf der X-Achse wird das Alter der Flugzeuge aufgetragen. In Abhängigkeit davon wird die Überlebenswahrscheinlichkeit der Flugzeuge auf der Y-Achse aufgetragen.

Die Spalten `Aircraft Age` und `Survival Propability` werden aus dem `DataFrame` der Eingabedaten sowie aus den vorhergesagten Daten für die Visualisierung extrahiert.

Das Bild des erzeugten Graphen kann als Portable Network Graphic (PNG) gesichert oder nach dem Joint Photographic Experts Group Standard (JPEG) exportiert werden.

8 Validierung

Nachdem die Neuronale Netze trainiert wurden, müssen sie auf ihre Glaubwürdigkeit überprüft werden. Dafür wird ein Vergleich der vorhergesagten Ergebnisse mit den Trainingsdaten durchgeführt. Darüber hinaus werden die entstandenen Retirement Curves mit anderen Vorhersagen abgeglichen.

Außerdem befasst sich dieses Kapitel ausführlicher mit dem in Abschnitt 5.4.2 beschriebenen Problem der Wahl zwischen festen Werten und Veränderungsraten für die Wirtschaftsdaten.

Vorweg muss angemerkt werden, dass sich die Validierung aufgrund der Art der Vorhersage als sehr schwierig erweist. Normalerweise kann bei der Klassifizierung der Prozentsatz von korrekt klassifizierten Datensätzen als Anhaltspunkt verwendet werden. Da bei dieser Vorhersage aber anhand von klassifizierten Daten eine Regression vorgenommen wird, ist die herkömmliche Validierung der Testdaten nicht möglich. Für die Validierung der vorhergesagten Ergebnisse wird daher der Kolmogorov-Smirnov-Test [24] verwendet. Dabei wird die maximale Abweichung zwischen den Kurven der vermuteten und der empirischen Wahrscheinlichkeit berechnet.

Um die Daten weiterhin validieren zu können, werden die Ergebnisse visualisiert und mit anderen Prognosen von Retirement Curves verglichen.

8.1 Vergleich mit den Trainingsdaten

Als ersten Schritt der Validierung wird das vorhergesagte Ergebnis des Neuronale Netzes mit den tatsächlichen Werten der Trainingsdaten verglichen.

Mithilfe einer in Python geschriebenen Funktion wird der Kolmogorov-Smirnov-Test auf die vorhergesagten Testdaten und die Trainingsdaten angewendet.

8.1 Vergleich mit den Trainingsdaten

Alter	Prognosewert	Trainingsdatenwert	absolute Diferenz	Maximum
1	0.998466	1.000000	0.001534	
5	0.997013	1.000000	0.002987	
10	0.992349	0.979198	0.013151	
15	0.955676	0.954918	0.000758	
20	0.819999	0.882540	0.062541	
25	0.460216	0.479339	0.019122	
28	0.219903	0.384615	0.164712	X
30	0.138487	0.206081	0.067595	
35	0.064198	0.075221	0.011024	
40	0.031720	0.027027	0.004693	
45	0.014882	0.013889	0.000993	
50	0.007589	0.007353	0.000236	
54	0.003953	0.000000	0.003953	

Tabelle 8.1: Kolmogorov-Smirnov-Test für Narrowbody Prognose

Tabelle 8.1 zeigt die Ergebnisse des Kolmogorov-Smirnov-Tests für die Prognose der Narrowbody Flugzeuge. Die größte Abweichung der beiden Kurven liegt bei 28 Jahren. Mit einem Signifikanzniveau α von 10%, liegt die Abweichung von 0,164712 aber im akzeptierten Rahmen von

$$\frac{1,224}{\sqrt{n}} \underset{\text{für } n=54}{=} 0,166565. \quad (8.1)$$

Der in Gleichung (8.1) berechnete Wert besagt, dass alle Werte, die um weniger als den berechneten Wert abweichen, bei einem Signifikanzniveau von 10% akzeptiert werden, also den Test bestehen.

Die Formel [15] kann verwendet werden, um den kritischen Wert näherungsweise zu bestimmen, sobald die Anzahl der Beobachtungswerte n größer als 35 ist. Für kleinere n liegen die kritischen Werte in Tabellen vor. [8]

Weiterhin werden die Trainingsdaten sowie die vorhergesagten Ergebnisse visualisiert.

8.1 Vergleich mit den Trainingsdaten

Abbildung 8.1 zeigt die Flugzeuge des Typs Boeing 747 in den Trainingsdaten. Im Graphen werden die Überlebenswahrscheinlichkeiten in Abhängigkeit des Alters der Flugzeuge aufgetragen.

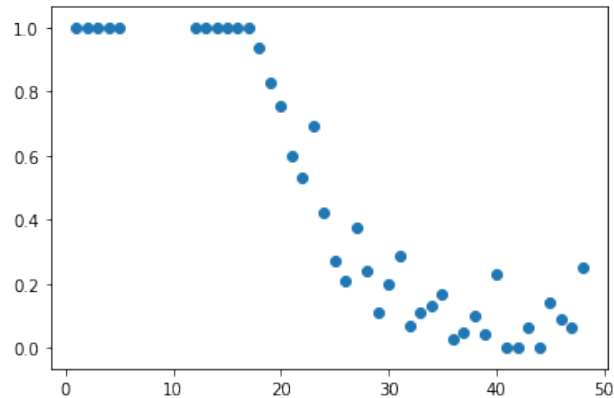


Abbildung 8.1: Flugzeuge des Typs Boeing 747 in den Trainingsdaten

Im Vergleich dazu zeigt Abbildung 8.2 die vorhergesagte Überlebenswahrscheinlichkeit der Flugzeuge. Beim Vergleichen der beiden Graphen wird deutlich, dass die Trainingsdaten nicht genau nachempfunden werden, sondern eine Kurve durch die Punktwolke gelegt wird. Eine exakte Kopie der Trainingsdaten ist nicht erwünscht, da dies für eine Überanpassung des Neuronalen Netzes an die Trainingsdaten sprechen würde.

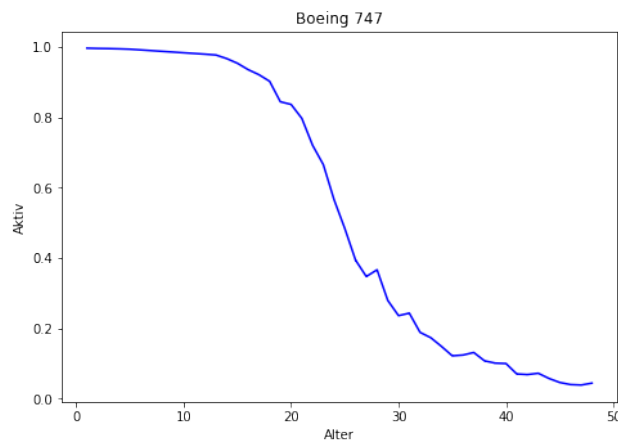


Abbildung 8.2: Vorhergesagte Überlebenswahrscheinlichkeit der Boeing 747

8.2 Vergleich mit anderen Vorhersagemodellen

Bei der Boeing 747 handelt es sich um ein Widebody Flugzeug. Ein weiterer Vergleich mit den Trainingsdaten kann mit Flugzeugen des Typs Narrowbody durchgeführt werden. Abbildung 8.3 zeigt den Vergleich der Narrowbody Trainingsdaten mit den Vorhersageergebnissen des Neuronalen Netzes.

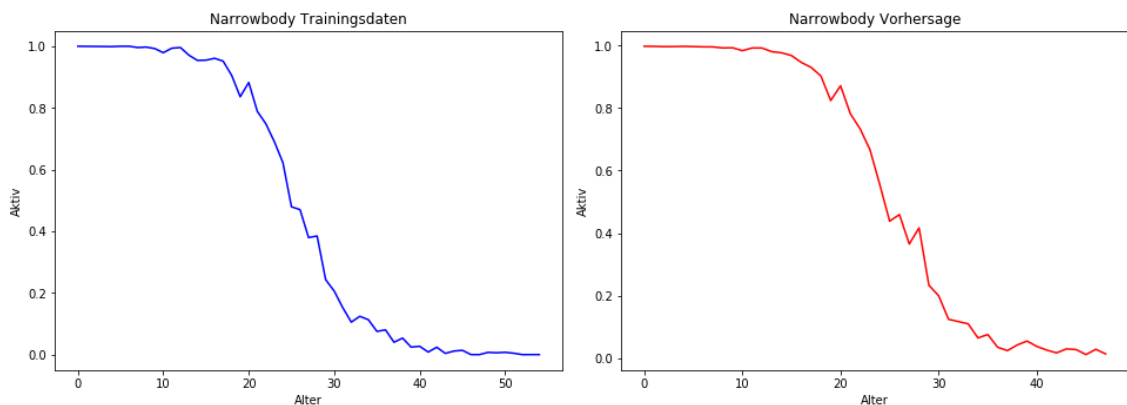


Abbildung 8.3: Vergleich von Narrowbody Trainingsdaten und Vorhersage

Die Vorhersage der Narrowbody Flugzeuge kommt den Trainingsdaten schon deutlich näher, jedoch werden nicht alle Peaks wie in den Trainingsdaten prognostiziert. Wenn überhaupt handelt es sich hier nur um eine leichte Form von Überanpassung. Der Gesamtverlauf der Retirement Curve ist logisch zu begründen: Je älter das Flugzeug ist, desto wahrscheinlicher wird es außer Betrieb genommen.

8.2 Vergleich mit anderen Vorhersagemodellen

In diesem Abschnitt werden die Ergebnisse der Vorhersage des Neuronalen Netzes mit anderen Vorhersagemodellen verglichen.

Abbildung 8.4 zeigt die Prognose von Retirement Curves des Instituts für Flughafenwesen und Luftverkehr des DLR aus dem Jahr 2017.

Der Kolmogorov-Smirnov-Test zeigt, dass die bestehende Vorhersage des DLR und die Prognose des Neuronalen Netzes ähnlich verlaufen. Tabelle 8.2 zeigt, dass die maximale Abweichung der beiden Retirement Curves bei 30 Jahren liegt und 0.165513 beträgt. Dieser Wert wird bei einem Signifikanzniveau α von 10% akzeptiert.

8.2 Vergleich mit anderen Vorhersagemodellen

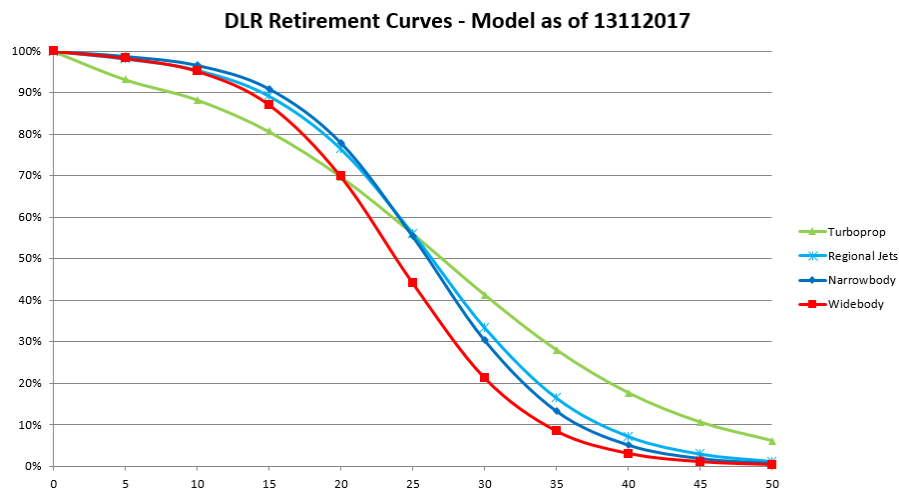


Abbildung 8.4: Retirement Curve Prognose von FW

Alter	Prognose NN	Prognose FW	absolute Differenz	Maximum
5	0.997013	0.988	0.009013	
10	0.992349	0.966	0.026349	
15	0.955676	0.910	0.045676	
20	0.819999	0.780	0.039999	
25	0.460216	0.554	0.093784	
30	0.138487	0.304	0.165513	X
35	0.064198	0.133	0.068802	
40	0.031720	0.051	0.019280	
45	0.014882	0.019	0.004118	
50	0.007589	0.007	0.000589	

Tabelle 8.2: Kolmogorov-Smirnov-Test: Vergleich Prognose FW und NN

Im Gegensatz zur in Abbildung 8.4 dargestellten Vorhersage von FW, wird in Abbildung 8.5 die Prognose des Neuronalen Netzes dargestellt.

Die Vorhersage des Neuronalen Netzes stimmt in weiten Teilen mit der Prognose von FW überein. Turboprop Flugzeuge halten länger als die anderen Modelle, wohingegen Regional Jets etwas schneller ausgemustert werden.

8.2 Vergleich mit anderen Vorhersagemodellen

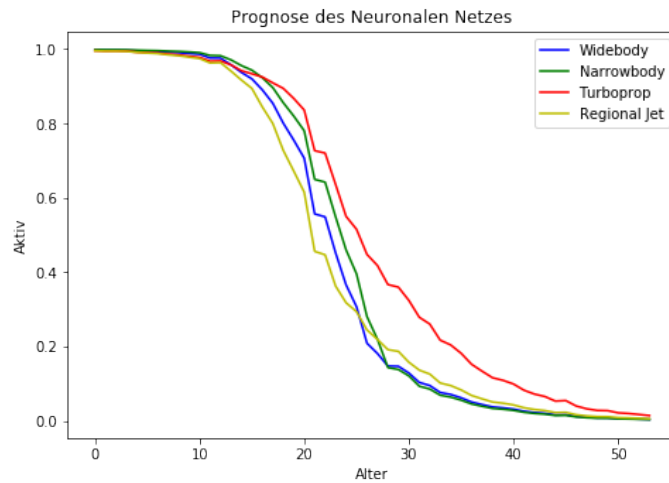


Abbildung 8.5: Retirement Curve Prognose des Neuronalen Netzes

Bei einem Blick auf die Trainingsdaten des Neuronalen Netzes (Abbildung 8.6) fällt auf, dass in den Trainingsdaten große Sprünge zwischen den einzelnen Jahren vorhanden sind. Dass das Neuronale Netz trotz dieser Trainingsdaten Retirement Curves produziert, die der Vorhersage von FW entsprechen, spricht für die gute Verallgemeinerung des NN. Durch eine größere und ausgeglicheneren Trainingsdatenbasis dürfte die Prognose des Neuronalen Netzes noch weiter verbessert werden. Jedoch ist das aktuelle Ergebnis bereits sehr zufriedenstellend.

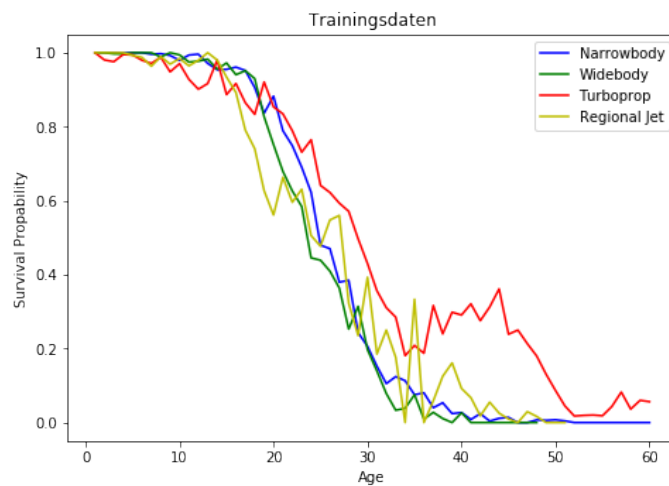


Abbildung 8.6: Visualisierung der Trainingsdaten

8.3 Scheinkorrelation durch Inflation

Schon bei ersten Tests während der Entwicklung ist aufgefallen, dass das Neuronale Netz eine starke Verbindung zwischen den ständig wachsenden Wirtschaftszahlen und der Überlebenswahrscheinlichkeit der Flugzeuge erkennt.

Das Wachstum des Bruttoinlandsprodukts ist aber zum Teil auf die stetige Inflation zurückzuführen. Die Korrelation zwischen Überlebenswahrscheinlichkeit und GDP ist somit eine Scheinkorrelation. Es wurde sich daher dafür entschieden, die Veränderungsraten der Wirtschaftsdaten im Vergleich zum Vorjahr zu verwenden anstelle der Wirtschaftsdaten an sich, um dieser Scheinkorrelation entgegenzuwirken.

Ein Vergleich der Vorhersagen zeigt, dass das Neuronale Netz, das mit den Veränderungsraten trainiert wurde, besser konvergiert und weniger anfällig für Fehler bei der Vorhersage ist.

9 Ergebnisse

Das Ergebnis dieser Arbeit ist ein Programm, das Retirement Curves für unterschiedliche Flugzeugtypen vorhersagen kann und dabei verschiedene Wirtschaftsszenarien simuliert.

Das Programm wird dabei aus vier Modulen aufgebaut. Hinzu kommt das einmalige Training der Neuronalen Netze.

Jedes der Module erfüllt eine spezielle Aufgabe im Programmablauf. Zuerst werden die Daten mit der Funktion `enrich_data` für die Verarbeitung im Neuronalen Netz vorbereitet. Dabei werden die Eingabedaten in eine Form gebracht, die vom Neuronalen Netz interpretiert werden kann. Außerdem wird die Umwandlung des nominalen Merkmals „Flugzeugtyp“ in mehrere binäre Spalten vorgenommen.

Die vorbereiteten Daten werden mit einer zusätzlichen Spalte versehen, in der das zu verwendende Modell für die Vorhersage notiert wird.

In der Funktion `process_dataframe` wird diese Spalte ausgelesen, damit das richtige Modell für die Vorhersage geladen werden kann.

Das Laden des Modells geschieht in der Funktion `load_scaler_and_model`. Dort wird das gewünschte Modell, welches per Parameter übergeben wird, von der Festplatte geladen und mit den ebenfalls abgespeicherten Gewichten initialisiert.

Daraufhin wird die Vorhersage der Daten durchgeführt. Diese prognostizierten Daten werden in der Funktion `plot_retirement_curve` visualisiert.

Die Ausgabe des Programms wird in Abbildung 9.1 dargestellt.

In der Abbildung ist zu erkennen, dass die Kurven einen zu anderen Vorhersagen ähnlichen Verlauf zeigen und die Trainingsdaten gut verallgemeinert wurden.

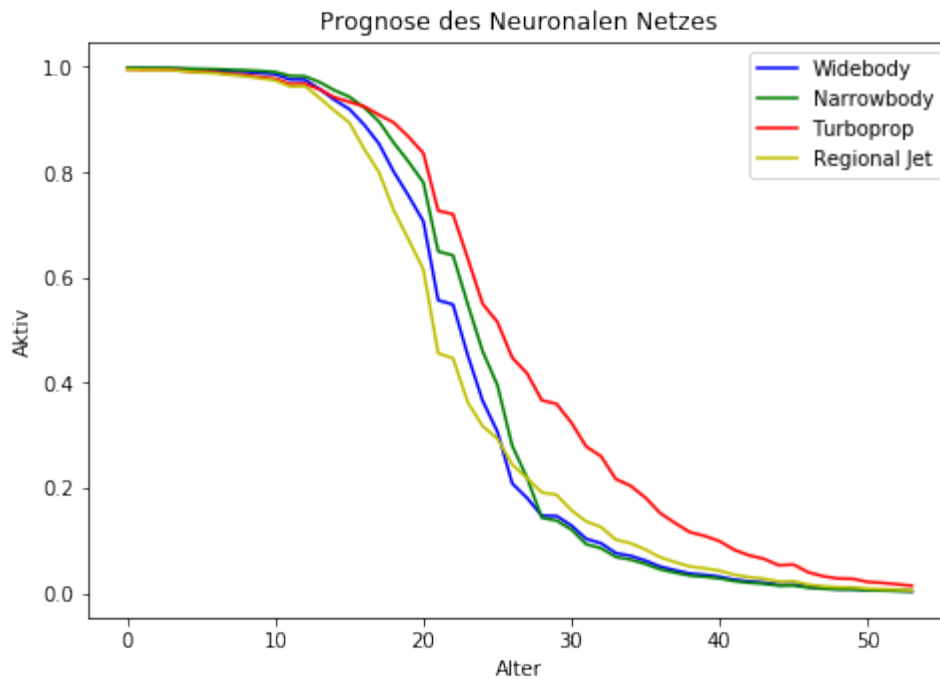


Abbildung 9.1: Ausgabe des Programms

Der große Knick beim Alter von 20 Jahren könnte eventuell mit der Jahrtausendwende und den damit verbundenen Computerumstellungen zusammenhängen. Eine Untersuchung der Wirtschaftsdaten zeigt jedoch keine starken Auffälligkeiten. Da das Neuronale Netz diesen Knick vorhersagt, sind in den Trainingsdaten wahrscheinlich für das menschliche Auge nicht sichtbare Zusammenhänge verborgen, die zu dieser Vorhersage führen.

In Tabelle 9.1 werden drei Wirtschaftsszenarien gezeigt.

Szenario	Entwicklung GDP	Entwicklung Ölpreis	Entwicklung Passagierzahlen
1	Low	Low	Low
2	Low	High	Low
3	High	High	High

Tabelle 9.1: Drei Wirtschaftsszenarien für den Vergleich

Die Vorhersage dieser Wirtschaftsszenarien wird in Abbildung 9.2 dargestellt. Diese Grafik zeigt, dass das Neuronale Netz für die verschiedenen Wirtschaftsszenarien abweichende Vorhersagen trifft. Einerseits kommt diese Differenz durch die unterschiedliche Entwicklung der Wirtschaftsdaten in jedem Szenario zustande, andererseits auch dadurch, dass das Training jedes Neuronalen Netzes getrennt voneinander abläuft. Somit ergibt sich für jedes Netz eine divergierende Prognose.

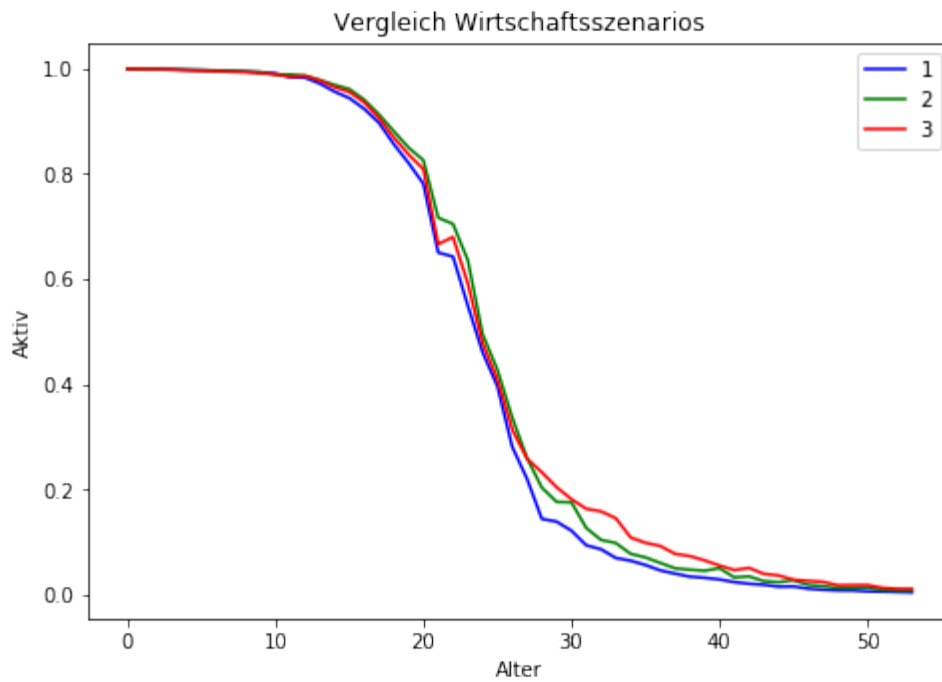


Abbildung 9.2: Vergleich verschiedener Wirtschaftsszenarien

10 Ausblick

In diesem Kapitel werden Erweiterungsmöglichkeiten für das entwickelte Programm vorgestellt. Darüber hinaus werden Verbesserungsmöglichkeiten beleuchtet und ein Ausblick auf zukünftige Arbeiten zum Thema Prognose von Retirement Curves mit Neuronalen Netzen gegeben.

Eine mögliche Verbesserung ist die Erweiterung der Trainingsdaten um das Kohortenjahr. Das Institut für Flughafenwesen und Luftverkehr des DLR hat damit bereits gute Erfahrungen bei der händischen Vorhersage von Retirement Curves gesammelt. Ebenso können die Trainingsdaten um weitere Wirtschaftsfaktoren erweitert werden, um so eventuelle Abhängigkeiten erkennen zu können.

Ein Beispiel dafür wäre die Aufgliederung der Passagierzahlen in Kurz- und Langstreckenflüge.

Für eine Verbesserung der Performance des Neuronalen Netzes kann eine größere Auswahl an Netzwerkformen miteinander verglichen werden.

Des Weiteren kann eine Dimensionsreduzierung zu einer Verbesserung der Vorhersage führen. Eine mögliche Herangehensweise wäre die Aufteilung der verschiedenen Flugzeugtypen in einzelne Netze. Dies bedeutet, dass pro Flugzeugtyp ein eigenes Neuronales Netz trainiert wird, um den Fokus beim Training auf die wichtigen Trainingsparameter zu legen.

Eine andere Möglichkeit wäre eine andere Codierung für die Flugzeugtypen. Aktuell werden diese mittels einer One-Hot-Codierung in das Neuronale Netz eingegeben. Eventuell würde eine Label-Codierung oder eine Dictionary-Codierung zu einer Performancesteigerung des Neuronalen Netzes führen. Das sind andere Formen der Codierung, die nicht für jeden Flugzeugtyp eine neue Spalte erzeugen, sondern ihn durch einen Zahlenwert ersetzen.

Die Dimensionsreduzierung könnte möglicherweise die Spitzen der vorhergesagten Retirement Curves etwas abflachen, da durch den „Fluch der Dimensionalität“, also eine dünne Verteilung von Daten auf viele Dimensionen, Overfitting gefördert wird. Aktuell gibt es aber kein deutliches Overfitting, sodass dies eine Möglichkeit der Bekämpfung von eventuellen Performanceverlusten während der Erweiterung wäre.

Insgesamt verhält sich die Vorhersage von Retirement Curves mithilfe Neuronaler Netze wie alle Machine Learning Herausforderungen:

Eine zu geringe Trainingsmenge sorgt für Probleme. Es ist davon auszugehen, dass eine Erweiterung der Trainingsdaten unweigerlich zu einer Verbesserung der Vorhersage mithilfe der Neuronalen Netze führt.

Durch das kontinuierliche Wachstum der globalen Flugzeugflotte in den nächsten Jahren kann von einer autonomen Steigerung der Performance des Neuronalen Netzes ausgegangen werden, da die Anzahl an Trainingsdaten mit jedem neu gebauten Flugzeug wächst.

Alles in allem ist die Vorhersage der Retirement Curves mithilfe des Neuronalen Netzes ein gelungener Ansatz zur Prognose der globalen Flugzeugflotte. Mit dieser Arbeit ist die Forschung auf diesem Gebiet jedoch bei weitem noch nicht abgeschlossen.

Es gibt viele Möglichkeiten der Optimierung, bei denen mit verschiedenen Parametern experimentiert werden muss, um herauszufinden, wie eine Verbesserung der Vorhersage erreicht werden kann.

Einige Ansätze wurden hier schon näher erläutert und können in Zukunft im Institut für Flughafenwesen und Luftverkehr genauer untersucht werden.

Literatur

- [1] Eingesehen am 07.07.2020. URL: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.
- [2] Martín Abadi u. a. „TensorFlow: A System for Large-Scale Machine Learning“. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, 11/2016, S. 265–283. URL: <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>.
- [3] Gerd Aberle. „Transportwirtschaft: einzelwirtschaftliche und gesamtwirtschaftliche Grundlagen“. In: Walter de Gruyter, 2013. Kap. 2.
- [4] *Alte Preise für Hardware*. Eingesehen am 06.07.2020. URL: <https://winhistory.de/more/386/altepreise.htm>.
- [5] *CAEP Forecast and Economic Analysis Support Group*. Eingesehen am 24.07.2020. URL: <https://www.icao.int/environmental-protection/Pages/CAEP-FESG.aspx>.
- [6] Hsun Chao u. a. „A model of aircraft retirement and acquisition decisions based on net present value calculations“. In: *17th AIAA Aviation Technology, Integration, and Operations Conference*. 2017, S. 3600.
- [7] *Cox Proportional-Hazards Model*. Eingesehen am 09.07.2020. URL: <http://www.sthda.com/english/wiki/cox-proportional-hazards-model>.
- [8] *Critical values for the Kolmogorov-Smirnov Test*. Eingesehen mit einem Webarchiv am 01.09.2020. URL: https://web.archive.org/web/20160818104718/http://www.mathematik.uni-kl.de/~schwaar/Exercises/Tabellen/table_kolmogorov.pdf.
- [9] Christoph Engelke. *Flottenplanung und -einsatz: Untersuchungen zur Weiterentwicklung eines dynamischen Flottenmodells für Leistungsanalysen der Weltflotte*. 2014.
- [10] Dick Forsberg. „Aircraft retirement trends & Outlook“. In: *Avolon, September* (2012).
- [11] Aurélien Géron. „Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow“. In: 2. Aufl. O’Reilly, 2019. Kap. 1.

- [12] Antoine Grall u. a. „Continuous-time predictive-maintenance scheduling for a deteriorating system“. In: *IEEE transactions on reliability* 51.2 (2002), S. 141–150.
- [13] Hashem M Hashemian. „State-of-the-art predictive maintenance techniques“. In: *IEEE Transactions on Instrumentation and measurement* 60.1 (2010), S. 226–236.
- [14] ROBERT HECHT-NIELSEN. „III.3 - Theory of the Backpropagation Neural Network**Based on “nonindent” by Robert Hecht-Nielsen, which appeared in Proceedings of the International Joint Conference on Neural Networks 1, 593–611, June 1989. © 1989 IEEE.“ In: *Neural Networks for Perception*. Hrsg. von Harry Wechsler. Academic Press, 1992, S. 65–93. DOI: <https://doi.org/10.1016/B978-0-12-741252-8.50010-8>. URL: <http://www.sciencedirect.com/science/article/pii/B9780127412528500108>.
- [15] Jürgen Hedderich und Lothar Sachs. *Angewandte Statistik*. Springer, 2016.
- [16] Aiwina Heng u. a. „Intelligent condition-based prediction of machinery reliability“. In: *Mechanical Systems and Signal Processing* 23.5 (2009), S. 1600–1614.
- [17] *ICAO Long-Term Traffic Forecasts*. Eingesehen am 24.07.2020. URL: <https://www.icao.int/safety/ngap/NGAP8%20Presentations/ICAO-Long-Term-Traffic-Forecasts-July-2016.pdf>.
- [18] *IEA World Energy Model (2019)*. Eingesehen am 16.07.2020. URL: <https://www.iea.org/reports/world-energy-model>.
- [19] Robert B Jackson u. a. „The environmental costs and benefits of fracking“. In: *Annual review of Environment and Resources* 39 (2014), S. 327–362.
- [20] Rahul Kar. „Dynamics of implementation of mitigating measures to reduce CO2 emissions from commercial aviation“. Diss. Massachusetts Institute of Technology, 2010.
- [21] *Keras API Documentation*. Eingesehen am 08.07.2020. URL: <https://keras.io/api/>.
- [22] *Keras Developer Guides*. Eingesehen am 08.07.2020. URL: <https://keras.io/guides/>.
- [23] *Keras Website*. Eingesehen am 08.07.2020. URL: <https://keras.io/>.
- [24] *Kolmogorov-Smirnov-Test*. Eingesehen am 01.09.2020. URL: <https://www.statistik-nachhilfe.de/ratgeber/statistik/induktive-statistik/signifikanztests-hypothesentests/pruefung-auf-verteilung-anpassungstests/kolmogorov-smirnov-test-liliefors-test>.

- [25] David Kriesel. *Ein kleiner Überblick über Neuronale Netze*. 2007. URL: [erh%5C%22%7Ba%7Dltlich%20auf%20http://www.dkriesel.com](http://www.dkriesel.com).
- [26] *Künstliche Intelligenz - Google Trends*. Eingesehen am 22.07.2020. URL: <https://trends.google.com/trends/explore?date=all&q=%2Fg%2F11c1r2rvnp, TensorFlow, Machine%20Learning, Neural%20Network, Deep%20Learning>.
- [27] Rocco Langone u. a. „Kernel spectral clustering for predicting maintenance of industrial machines“. In: *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. IEEE. 2013, S. 39–45.
- [28] *Machine Learning on Big Data*. Eingesehen am 02.07.2020. URL: https://www.ebtic.org/source/ebtics_view_ml_big_data/image001.jpg.
- [29] *Neuronale Netze - Eine Einführung*. Eingesehen am 08.07.2020. URL: <http://www.neuronaletesnetz.de/backpropagation2.html>.
- [30] Xander Olsthoorn. „Carbon dioxide emissions from international aviation: 1950–2050“. In: *Journal of Air Transport Management* 7.2 (2001), S. 87–93. DOI: [https://doi.org/10.1016/S0969-6997\(00\)00031-4](https://doi.org/10.1016/S0969-6997(00)00031-4). URL: <http://www.sciencedirect.com/science/article/pii/S0969699700000314>.
- [31] *Orders & Deliveries*. Eingesehen am 09.07.2020. URL: <http://www.boeing.com/commercial/#/orders-deliveries>.
- [32] *Predictive Maintenance*. Eingesehen am 22.07.2020. URL: <https://www.bigdata-insider.de/was-ist-predictive-maintenance-a-640755/>.
- [33] *Python object serialization*. Eingesehen am 29.07.2020. URL: <https://docs.python.org/3/library/pickle.html>.
- [34] Andrinandrasana David Rasamoelina, Fouzia Adjailia und Peter Sincak. „A Review of Activation Function for Artificial Neural Network.“ In: *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI), Applied Machine Intelligence and Informatics (SAMI), 2020 IEEE 18th World Symposium on* (2020), S. 281–286. URL: <https://login-ezproxy-dhma-redi-bw-de.ezproxy-dhma-1.redi-bw.de/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsee&AN=edsee.9108717&lang=de&site=eds-live>.
- [35] Sebastian Raschka. „Machine Learning mit Python“. In: mitp Verlags GmbH & Co. KG, 2017. Kap. 1.2.
- [36] Sebastian Raschka. „Machine Learning mit Python“. In: mitp Verlags GmbH & Co. KG, 2017. Kap. 2.
- [37] Tariq Rashid. *Neurale Netze selbst programmieren*. 1. Aufl. O’Reilly, 2017.

- [38] Martin Schaefer. „Development of forecast model for global air traffic emissions“. Diss. DLR, 2012, S. 71–74.
- [39] *Skizze des Tectums eines Spatzen*. Eingesehen am 07.07.2020. URL: <https://commons.wikimedia.org/wiki/File:SparrowTectum.jpg>.
- [40] *Stooq - Historical Data: Crude Oil WTI - NYMEX (CL.F)*. Eingesehen am 30.06.2020. URL: <https://stooq.com/q/d/?s=cl.f&i=y&l=2>.
- [41] Sanjiv Das Subir Varma. „Deep Learning“. In: 2018. Kap. 4. URL: <https://srdas.github.io/DLBook/LinearLearningModels.html>.
- [42] Gian Antonio Susto u. a. „Machine learning for predictive maintenance: A multiple classifier approach“. In: *IEEE Transactions on Industrial Informatics* 11.3 (2014), S. 812–820.
- [43] *SystemX, Institute de Recherche Technologique*. Eingesehen am 14.03.2020. URL: <https://blog.irt-systemx.fr/machine-learning-et-conduite-autonome>.
- [44] *Tecson - Entwicklung der Erdölpreise*. Eingesehen am 30.06.2020. URL: <https://www.tecson.de/historische-oelpreise.html>.
- [45] *TensorFlow Lizenz*. Eingesehen am 08.07.2020. URL: <https://github.com/tensorflow/tensorflow/blob/master/LICENSE>.
- [46] *TensorFlow Website*. Eingesehen am 08.07.2020. URL: <https://www.tensorflow.org/about>.
- [47] *THE WORLD BANK - GDP per capita*. Eingesehen am 29.06.2020. URL: <https://data.worldbank.org/indicator/NY.GDP.PCAP.CD?end=2018&start=1960>.
- [48] *Universitätsbibliothek Humboldt-Universität zu Berlin*. Eingesehen am 06.07.2020. URL: <https://www.digi-hub.de/viewer/image/BV041097321/65/>.
- [49] Kathleen Vaillancourt u. a. „A Canadian 2050 energy outlook: Analysis with the multi-regional model TIMES-Canada“. In: *Applied Energy* 132 (2014), S. 56–65. DOI: <https://doi.org/10.1016/j.apenergy.2014.06.072>. URL: <http://www.sciencedirect.com/science/article/pii/S0306261914006540>.
- [50] Sebastian Erik Vock. *FLOTTEPLANUNG VON FLUGZEUGEN*. 2010.
- [51] *Was sind künstliche neuronale Netze?* Eingesehen am 07.07.2020. URL: <https://rocketloop.de/kuenstliche-neuronale-netze/>.
- [52] Michael Wetzer. *Performing predictive maintenance on equipment*. US Patent 6,738,748. 05/2004.

- [53] Jiawei Zhang. *Basic Neural Units of the Brain: Neurons, Synapses and Action Potential*. 2019. arXiv: 1906.01703 [q-bio.NC].
- [54] Andreas Ziegler, S Lange und Ralf Bender. „Überlebenszeitanalyse: Die Cox-Regression“. In: *DMW-Deutsche Medizinische Wochenschrift* 132.S 01 (2007), e42–e44.