



TECHNISCHE
UNIVERSITÄT
WIEN

DIPLOMARBEIT

Optimale Rückversicherungsstrategien für Langlebigkeitsrisiko mittels Deep Learning

ausgeführt am

Institut für
Stochastik und Wirtschaftsmathematik
TU Wien

unter der Anleitung von

Univ.Prof. Dipl.-Math. Dr.rer.nat Thorsten Rheinländer

durch

Viktoria Katzensteiner, BSc

Matrikelnummer: 1425723

Wien, am 4. September 2022

Kurzfassung

In dieser Arbeit wird eine relativ neue Herangehensweise erläutert, mit der sich Versicherungsunternehmen gegen das Langlebkeitsrisiko absichern können. Das Risiko besteht darin, dass Versicherungsnehmer von Rentenprodukten länger leben als erwartet und Renten somit länger vom Versicherer ausbezahlt werden müssen. Da sich Finanzinstrumente wie Longevity Swaps zur Absicherung gegen dieses Risiko auf dem Markt nicht so durchgesetzt haben, wie dies erwartet wurde, versuchen Versicherer wieder vermehrt dieses Risiko mittels Rückversicherung abzusichern. Im Zeitalter von Big Data gewinnt das Thema Deep Learning immer mehr an Bedeutung, weshalb in dieser Arbeit versucht wird den Rückversicherungsbetrag mittels Neuronaler Netze vorherzusagen. Es ist sehr schwierig zukünftige Mortalitätstrends zu prognostizieren, daher werden möglichst viele verschiedene Annahmen an die Mortalität in das Modell miteinbezogen. Eine Möglichkeit dies zu erreichen ist, indem man das Modell mit Daten trainiert, welche stochastische Komponenten enthalten. Diese stochastischen Komponenten sollen das Langlebkeitsrisiko widerspiegeln. Im Trainingsprozess werden die Parameter und Gewichte des Modells so gewählt, dass der Fehler zwischen vorhergesagten und tatsächlichen Werten minimal ist. Somit erhält man am Ende ein Modell, welches die Höhe der Rückversicherung unter verschiedensten Annahmen an die Mortalität mit einem tolerierten Fehler prognostiziert.

Abstract

In this thesis, we take a look at a relatively new approach on how insurance companies can secure themselves against longevity risk. The longevity risk is the risk that policyholders of annuity products live longer than expected. This results in longer annuity payments for the insurer. There are a couple of financial instruments like longevity swaps for securing against this risk on the market, but none of them are as established as they were expected to be. Thus, more and more insurers try to hedge their longevity risk with reinsurance. Deep Learning is becoming more important in times of Big Data, therefore we try to predict the reinsurance amount with Neural Networks. It is very difficult to predict future mortality trends, thus we want the model to include many different assumptions about mortality. We achieve this by training the model with data that contains stochastic components. These stochastic components should represent the longevity risk. The training process of a Neural Network consists of adapting weights and parameters in a way that the loss between predicted and actual values is minimized. In the end, we obtain a model that covers many different assumptions about mortality and predicts the reinsurance value with a tolerable loss.

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich auf meinem Weg durchs Studium und während der Anfertigung meiner Diplomarbeit unterstützt und motiviert haben.

Zuerst möchte ich mich bei Herrn Professor Univ.Prof. Dipl.-Math. Dr.rer.nat Thorsten Rheinländer bedanken für das spannende Thema und die überaus gute Betreuung. Ich konnte mir in dieser Zeit sehr viel neues und wertvolles Wissen anlernen.

Ebenfalls möchte ich mich bei meinen Studienkollegen bedanken, die mittlerweile sehr gute Freunde geworden sind. Danke für die große Unterstützung und Motivation weiter zu machen, auch wenn es manchmal schwierig war.

Einen großen Dank möchte ich auch meiner Familie aussprechen, die mir dieses Studium ermöglicht hat und mir immer gut zugesprochen hat. Danke für eure Unterstützung in vielerlei Hinsicht.

Und zu guter Letzt noch ein großes Dankeschön an meinen Freund, der mich fast durch meine gesamte Studienzzeit begleiten durfte und mich immer motiviert hat mein Bestes zu geben. Danke für deine Geduld und die großartige Unterstützung.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 4. September 2022

Viktoria Katzensteiner, BSc

Inhaltsverzeichnis

1	Einleitung	1
2	Langlebigkeitsrisiko	3
2.1	Schwierigkeiten bei der Vorhersage von Langlebigkeitstrends	3
2.2	Modelle für die Vorhersage von Mortalitätstrends	4
2.2.1	Lee-Carter Modell	5
2.3	Rückversicherung des Langlebigkeitsrisikos	7
2.3.1	Basisrisiko und Gegenparteiausfallrisiko	8
2.3.2	Rückversicherungsvereinbarungen	9
3	Rückversicherung	12
3.1	Allgemeines und Definitionen	12
3.2	Formen der Rückversicherung	14
3.2.1	Die fakultative Rückversicherung	15
3.2.2	Die obligatorische Rückversicherung	16
3.3	Proportionale Rückversicherung	17
3.3.1	Summenexzedentenrückversicherung	18
3.3.2	Quotenrückversicherung	19
3.4	Nicht-proportionale Rückversicherung	20
3.4.1	Schadenexzedentenrückversicherung	22
3.4.2	Stop-Loss Rückversicherung	23
3.4.3	Prämienberechnung	23
4	Deep Learning	28
4.1	Grundlagen Machine Learning	28
4.1.1	Lernende Algorithmen	28
4.1.2	Kapazität, Overfitting und Underfitting	30
4.1.3	Hyperparameter und Validierungsdatensatz	31
4.1.4	Motivation für Deep Learning	31
4.2	Aufbau Neuronaler Netzwerke	31
4.2.1	Das Perzeptron	32
4.2.2	Feedforward Neuronale Netzwerke	33
4.3	Aktivierungsfunktionen	35
4.4	Verlustfunktionen	37
4.5	Optimierung Neuronaler Netzwerke	39
4.5.1	Backpropagation	39
4.5.2	(Stochastisches) Gradienten Abstiegsverfahren	41
4.5.3	Lernrate	42

4.5.4	Momentum	43
4.5.5	Algorithmen mit adaptiver Lernrate	44
5	Beispiel	47
5.1	Datensatzerstellung	47
5.2	Deep Learning Modelle	54
5.2.1	Tanh Modelle	54
5.2.2	ReLU Modelle	56
5.2.3	Vorhersage	58
	Abbildungsverzeichnis	60
	Literaturverzeichnis	61

1 Einleitung

Die vorliegende Arbeit handelt von optimalen Rückversicherungsstrategien für Langlebighkeitsrisiko mittels Deep Learning und ist daher in drei größere Themengebiete eingeteilt, nämlich Langlebighkeitsrisiko, Rückversicherung und Deep Learning. Zum Schluss folgt ein Beispiel, welches die Anwendung der Theorie zeigen soll.

Zu Beginn in Kapitel 2 wird erklärt, was das Langlebighkeitsrisiko allgemein ist und weshalb die Vorhersage der Mortalitäts- und Langlebighkeitstrends so schwierig ist. Weiters werden Modelle für die Vorhersage von Mortalitätsstrends aus [P⁺02] vorgestellt und auf das Lee-Carter Modell, welches in der Praxis Anwendung findet, etwas genauer eingegangen. Zum Schluss des zweiten Kapitels werden die möglichen Rückversicherungsvereinbarungen für das Langlebighkeitsrisiko und die damit verbundenen Risiken, das Basisrisiko und das Gegenparteiausfallsrisiko, erläutert.

Kapitel 3 handelt von der Rückversicherung und startet mit einigen grundlegenden Begriffen und Definitionen. Anschließend folgen die verschiedenen Formen der Rückversicherung. Zuerst werden die juristischen Formen der Rückversicherung erklärt, nämlich die fakultative und die obligatorische Rückversicherung. Danach folgt die Unterscheidung der technischen Formen in proportionale und nicht-proportionale Rückversicherung. Die Formen der proportionalen Rückversicherung sind wiederum die Summenexzedenten- und die Quotenrückversicherung. Die Formen der nicht-proportionalen Rückversicherung sind die Schadenexzedenten- und die Stop-Loss-Rückversicherung. Am Ende dieses Kapitels wird auf die Prämienberechnung der verschiedenen Rückversicherungsformen eingegangen.

Das Kapitel 4 Deep Learning erläutert zu Beginn die wichtigsten Grundprinzipien des Machine Learnings, da diese die Grundlage des Deep Learnings bilden. Danach wird der Aufbau Neuronaler Netzwerke beschrieben, startend mit dem Perzeptron. Nach der Definition eines Feedforward Neuronalen Netzwerks wird kurz auf wichtige Bestandteile wie Aktivierungs- und Verlustfunktionen eingegangen. Wenn das Grundgerüst des Netzwerkes steht, ist der nächste wichtige Schritt die Optimierung. Dafür wird einerseits die Backpropagation und andererseits das (stochastische) Gradientenverfahren vorgestellt. Dabei gilt, dass die Wahl der Lernrate sehr bedeutend ist. Da diese nicht konstant gewählt werden sollte, verwendet man für die Optimierung Algorithmen mit adaptiver Lernrate. Auf diese wird zum Schluss des Kapitels eingegangen.

Kapitel 5 startet mit der Generierung eines Datensatzes aus einem stochastischen Gompertz-Makeham Modell. Dieser Datensatz enthält zwei stochastische Komponenten, die das Langlebighkeitsrisiko darstellen. Mittels diesem wird das Deep Learning Modell trainiert. Ziel ist es unter verschiedenen Annahmen an die Mortalität den Betrag der Rückversicherung mit-

tels eines Neuronalen Netzwerks vorherzusagen. Dabei wird mit einem Modell bestehend aus 3 hidden Layern mit jeweils 64 Knoten, Tangens hyperbolicus als Aktivierungsfunktion und Adam als Optimizer gestartet. Dieses Modell wird verbessert, indem einige Hyperparameter abgeändert werden. Abschließend werden die Modelle herangezogen, um mit realen Daten der „AVÖ 2018-P - Rechnungsgrundlagen für Pensionsversicherung“ Vorhersagen zu machen.

2 Langlebighkeitsrisiko

Allgemein ist das Langlebighkeitsrisiko ein Risiko, dem Versicherungsunternehmen, die Langlebighkeitsprodukte anbieten, ausgesetzt sind. Das Risiko besteht darin, dass Versicherungsnehmer einer Altersvorsorge oder Rentenempfänger länger leben als erwartet und die Versicherer somit die Renten länger ausbezahlen müssen. Die Lebensdauer der Menschen verlängert sich aufgrund von externen Faktoren wie gesündere Lebensstile, ständige Verbesserung der medizinischen Versorgung und Durchbrüche in der Medizinwissenschaft. Versicherungsunternehmen wollen das Risiko, dem sie durch die Langlebighkeit ausgesetzt sind, möglichst verringern, da sie sonst sehr hohe Kapitalanforderungen haben. [CFvBM20]

Versicherungsunternehmen sind dem Langlebighkeitsrisiko durch Renten, Pensionen und Langzeitpflege ausgesetzt. Während sich der private Versicherungsmarkt für solche Produkte in manchen Ländern noch nicht sehr durchgesetzt hat, gibt es wiederum in anderen Ländern wie z.B. in Kanada, Großbritannien und den USA einen signifikanten Marktdurchbruch von Pensions- und Rentenprodukten.

Kurz gesagt, die Nachfrage für Langlebighkeitsprodukte wächst schnell. Auch wenn Versicherungsunternehmen darin eine enorme Geschäftschance sehen, hat das Risikomanagement solcher Produkte mit beachtlichen Schwierigkeiten zu kämpfen.

Über die letzten Jahrzehnte konnte man beobachten, dass die Lebenserwartung sowohl in den industrialisierten als auch in den weniger entwickelten Ländern gestiegen ist. Es wird angenommen, dass Langlebighkeit mehr als nur ein „neuer demographischer Trend“ ist, vielmehr verursacht es einen neuen demographischen Übergang. Da unter anderem die Geburtenrate deutlich gesunken ist, hat sich die allgemeine Altersstruktur der Bevölkerung stark verändert.[RHT05]

Pittaco [P+02] hat die Entwicklungen folgendermaßen zusammengefasst:

1. Ein Anstieg der Lebenserwartung (sowohl bei der Geburt als auch im Alter)
2. Ein genereller Anstieg des voraussichtlichen Todesalters
3. Eine Abnahme der Mortalitätsraten im Erwachsenenalter und hohem Alter

Da die zukünftigen Mortalitätstrends zufällig und somit schwer vorherzusagen sind, stellt dies ein Hindernis für Rückversicherungslösungen dar.

2.1 Schwierigkeiten bei der Vorhersage von Langlebighkeitstrends

Welche Schwierigkeiten die Vorhersage der Mortalitäts- bzw. Langlebighkeitstrends mit sich bringt wird aus [RHT05] entnommen.

Mittlerweile steht fest, dass Fortschritte in Ernährung, Hygiene und medizinischer Versorgung einen positiven Einfluss auf Lebenserwartungstrends haben. Viele Wissenschaftler erwarten, dass Fortschritte, speziell in der Gentechnik, einen Sprung in der Lebenserwartung auslösen könnten. Dies wiederum könnte dazu führen, dass Lebensversicherer ihre Reserven deutlich erhöhen müssen. Andererseits können Epidemien dazu führen, dass die Mortalitätsraten, zumindest kurzzeitig, steigen. Dies wäre zum Beispiel momentan der Fall aufgrund der Covid-19 Pandemie.

Ein Faktor, der die Vorhersage erschwert ist, dass Langlebigkeitsprodukte generell eine relativ lange Laufzeit haben. Vorhersagen über die Entwicklung über einen langen Zeitraum sind an sich schon sehr schwierig. Weiters ist der Markt für solche Produkte teilweise nicht sehr groß und da das Erstellen von Sterbetafeln schwierig und teuer ist, gibt es in vielen Ländern keine Tafeln für diese Produkte.

Ein weiter Grund ist, dass sich die versicherte Gruppe auch wesentlich von der Gesamtbevölkerung unterscheiden kann.

2.2 Modelle für die Vorhersage von Mortalitätstrends

Für die grundlegende Idee und damit verbundenen Risiken der Modelle wird auf [P⁺02] verwiesen.

Die neuesten Mortalitätstrends führen dazu, prognostizierte Survivalmodelle für die Bepreisung und für die Aufstellung von Reserven für Lebensrenten und anderen langfristigen Rentenprodukten zu verwenden.

Die Grundidee für die Vorhersage der Mortalität ist, die Mortalität selbst als Funktion eines zukünftigen Kalenderjahres y zu sehen, also beispielsweise die reellwertige Funktion $\Psi(y)$. Für aktuarielle Berechnungen muss die Funktion auch von einem Alter x abhängen. Dann ist die Mortalität für jedes Alter x eine Funktion des Kalenderjahres y , also eine Funktion $\Psi(x, y)$.

Solch eine Funktion wird konstruiert, indem man passende statistische Verfahren auf die vergangene Mortalitätserfahrung anwendet. Allgemein kann die Funktion Ψ Mortalitätsraten, Mortalitätswahrscheinlichkeiten, eine Survivalfunktion, eine Transformation der Survivalfunktion, usw. darstellen.

Modelle für Vorhersagen erhält man, indem Extrapolationsverfahren auf die beobachtete Mortalität der Vergangenheit angewandt werden. Dies wird in Abbildung 2.1 veranschaulicht.

Aufgrund zufälliger Schwankungen in den beobachteten Mortalitätsraten kann es zu Abweichungen von den vorhergesagten Mortalitätsraten kommen. Das Risiko der zufälligen Schwankungen ist ein bekanntes Risiko in der Versicherungsbranche. Dieses Risiko kann

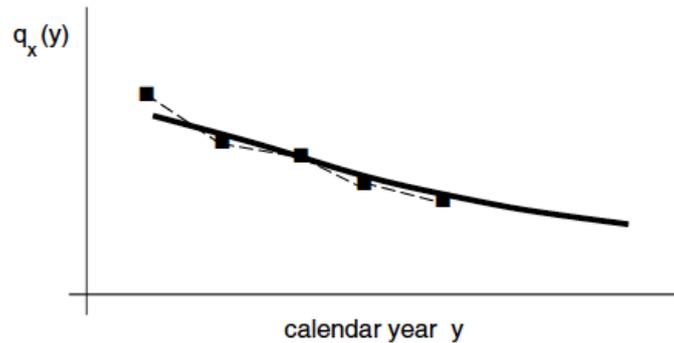


Abbildung 2.1: Extrapolation of the mortality profile
Quelle: Ermanno Pitacco et al., 2002, Seite 8

jedoch mit zunehmender Portfoliogröße verringert werden. Daher wird es auch „Pooling Risiko“ genannt.

Andererseits gibt es den Fall, dass Abweichungen nicht nur aufgrund zufälliger Schwankungen vorkommen, sondern dadurch entstehen, dass der tatsächliche Mortalitätstrend vom vorhergesagten abweicht. So entstehen systematische Abweichungen. Dieses Risiko kann allerdings nicht durch Vergrößern des Portfolios gehedgt werden. Aus diesem Grund wird das systematische Abweichungsrisiko auch „Non-pooling Risiko“ genannt.

Es ist somit wichtig zu wissen, dass zukünftige Mortalitätstrends zufällig sind. Daher macht es keinen Unterschied welches Verfahren für die Vorhersage verwendet wird, es können immer systematische Abweichungen auftreten.

Es gibt mittlerweile einige Modelle für Vorhersagen, die auch in der Praxis Anwendung finden. Ein sehr bekanntes Modell ist das Lee-Carter Modell, welches Zufälligkeit in den zukünftigen Mortalitätstrends zulässt.

2.2.1 Lee-Carter Modell

Dieser Abschnitt basiert auf [Lee00].

Lee und Carter haben eine Methode für die Extrapolation von Mortalitätstrends und Altersstrukturen entwickelt. Diese Methode ist für langfristige Vorhersagen geeignet. Einerseits hat sie einige Vorteile gegenüber anderen Extrapolationsmethoden, nichtsdestotrotz teilt sie auch die allgemeinen Schwächen der Extrapolation. Diese wären unter anderem, dass historische Muster nicht immer für die Zukunft gelten müssen und dadurch strukturelle Veränderungen übersehen werden können. Es werden keine neuen Erkenntnisse über Fortschritte in Medizin, verändernde Lebensstile oder neue Krankheiten berücksichtigt.

Prinzipiell verlangsamt sich die Zunahme der Lebenserwartung in vielen Bevölkerungen.

Diese Eigenschaft muss für plausible Vorhersagen erfüllt sein. Um diese „Verlangsamung bzw. Verzögerung“ in ein Vorhersagemodell zu integrieren, kann zum Beispiel ein oberes Limit für die Lebenserwartung eingeführt werden.

Das Modell

Für die Beschreibung des Modells werden zusätzliche Informationen aus [LC92] und [LM01] entnommen.

Das Lee-Carter Modell für die Mortalität hat folgende Darstellung:

$$\ln(\mathbf{m}_{x,t}) = \mathbf{a}_x + \mathbf{b}_x \mathbf{k}_t + e_{x,t}, \quad (2.1)$$

wobei

- $\mathbf{M} \in \mathbb{R}^{n \times n}$ die Matrix der Sterberaten ist und die Einträge $\mathbf{m}_{x,t}$ die zentralen Sterberaten für eine x -jährige Person zur Zeit t sind
- $\mathbf{a} \in \mathbb{R}^n$, wobei \mathbf{a}_x die durchschnittliche Struktur der Altersgruppe x beschreibt
- $\mathbf{k} \in \mathbb{R}^n$, wobei \mathbf{k}_t ein Index für das allgemeine Mortalitätsniveau zur Zeit t ist
- $\mathbf{b} \in \mathbb{R}^n$, wobei \mathbf{b}_x die Entwicklung der Mortalität zum Alter x bei änderndem Parameter \mathbf{k}_t beschreibt
- $e_{x,t}$ ein Fehlerterm ist

für $n \in \mathbb{N}$.

Durch die Modellierung des Logarithmus von $m_{x,t}$ wird sichergestellt, dass die altersspezifischen Sterberaten nie negativ werden. Dies stellt einen erheblichen Vorteil des Modells dar.

Die exponentielle Änderungsrate der Mortalität einer Altersgruppe ist proportional zu \mathbf{b}_x . Der Term \mathbf{b}_x verrät uns somit, welche Raten schnell und welche langsam abfallen aufgrund von Änderungen in \mathbf{k} , da gilt:

$$\frac{d \ln(\mathbf{m}_{x,t})}{dt} = \mathbf{b}_x \frac{d\mathbf{k}_t}{dt}. \quad (2.2)$$

Der Fehlerterm $e_{x,t}$ spiegelt altersspezifische Einflüsse wider, die nicht ins Modell einbezogen wurden, wie zum Beispiel die oben genannten Fortschritte in der Medizin. Man hofft, dass sich der Fehlerterm „brav“ verhält und eine relativ kleine Varianz aufweist.

Fitting des Modells

Auch dieser Abschnitt beruht zusätzlich auf [LC92] und [LM01].

Zuerst müssen folgende Annahmen getroffen werden, damit die daraus resultierende Lösung eindeutig ist:

- Die Summe über alle \mathbf{b}_x ist gleich Eins.
- Die Summe über alle \mathbf{k}_t ist gleich Null.

Daraus ergibt sich, dass \mathbf{a}_x genau den durchschnittlichen Werten der $\ln(m_{x,t})$ für jedes x über einen längeren Zeitraum $t \in \{1, \dots, T\}$ entspricht, also:

$$\mathbf{a}_x = \frac{1}{T} \sum_{t=1}^T \ln(m_{x,t}). \quad (2.3)$$

Da alle Werte auf der rechten Seite der Modellgleichung (2.1) unbekannt sind, kann das Fitting nicht mittels linearer Regression geschehen.

Das Modell wird dann folgendermaßen an historische Daten gefittet.

Die Werte für \mathbf{a}_x , \mathbf{b}_x und \mathbf{k}_t können mittels Anwendung der Singulärwertzerlegung auf die Matrix \mathbf{M} ermittelt werden, die entsteht, wenn man von den logarithmierten Sterberaten $\ln(m_{x,t})$ die durchschnittlichen altersspezifischen Raten \mathbf{a}_x abzieht.

Dieses Verfahren liefert eine Lösung mittels der Methode der kleinsten Quadrate.

Die Werte von \mathbf{k}_t bilden eine Zeitreihe. Statistische Standardmethoden können dann verwendet werden, um diese Zeitreihe zu modellieren und vorherzusagen. Das vorhergesagte \mathbf{k}_{t+n} und die geschätzten Werte für \mathbf{a}_x und \mathbf{b}_x können dann in die Modellgleichung (2.1) eingesetzt werden, um Vorhersagen für altersspezifische Sterberaten $m_{x,t+n}$ zu treffen.

2.3 Rückversicherung des Langlebigkeitsrisikos

Grundsätzliches über die Rückversicherung des Langlebigkeitsrisikos und die wiederum damit verbundenen Risiken basiert auf [CFvBM20].

Das Langlebigkeitsrisiko wird oft zwischen zwei Parteien mittels Rückversicherung oder Langlebigkeitsswaps übertragen. Eine wichtige Grundvoraussetzung einer solchen Übertragung ist, dass es Parteien am Rückversicherungsmarkt gibt, die groß genug sind, um dieses Risiko zu absorbieren. In dieser Arbeit wird nur auf die Rückversicherungsvereinbarungen eingegangen.

Das Langlebigkeitsrisiko kann gerade für Pensionskassen und Pensionsversicherer erheblich sein. Ein Hauptgrund, warum Versicherer dieses Risiko möglichst reduzieren wollen ist, dass es hohe Kapitalanforderungen widerspiegelt.

Das Langlebigkeitsrisiko kann einen natürlichen Hedge mit dem Mortalitätsrisiko bilden. Versicherer und Rückversicherer können daher das Langlebigkeitsrisiko als ausgleichenden Hedge für das Mortalitätsrisiko verwenden und umgekehrt. Folglich kann die Übertragung des Langlebigkeitsrisikos für beide Parteien von Vorteil sein. Falls beispielsweise der Versicherer hauptsächlich dem Langlebigkeitsrisiko ausgesetzt ist und der Rückversicherer

hauptsächlich dem Mortalitätsrisiko, dann entsteht durch den teilweisen Transfer des Langlebigkeitsrisikos vom Versicherer an den Rückversicherer ein ausgewogeneres Risikoprofil für beide Parteien, was zu geringeren Kapitalanforderungen führt.

2.3.1 Basisrisiko und Gegenparteiausfallrisiko

Die Verwendung von Rückversicherungsvereinbarungen für das Langlebigkeitsrisiko bringt jedoch neue Risiken mit sich. Diese wären das Basisrisiko und das Gegenparteiausfallrisiko, die man wiederum auch möglichst gering halten will.

Das *Basisrisiko* ist jenes Risiko, das entsteht, falls der Rückversicherungsschutz einen unzureichenden Hedge des zu transferierenden Risikos darstellt. Falls der Versicherer zum Beispiel dem Mortalitätsrisiko in seinem Portfolio ausgesetzt ist und dieses Risiko mittels Mortalitätsbonds hedgen will, dann entsteht das Basisrisiko dieses Hedges zu dem Ausmaß, zu dem der Mortalitätsindex des Mortalitätsbonds von der Mortalität des Portfolios abweicht.

Das *Gegenparteiausfallrisiko* ist verbunden mit der möglichen Zahlungsunfähigkeit des Rückversicherers und hat folgende Auswirkungen für den Versicherer unter Solvency II. Einerseits besteht die Möglichkeit, dass der Versicherer nicht alle zukünftigen einforderbaren Beträge aus der Rückversicherung erhält und dies somit Auswirkungen auf die Eigenmittel des Versicherers hat. Andererseits kann dieses Ausfallrisiko die Effektivität des Rückversicherungsschutzes in der Berechnung des benötigten Kapitals und der Eigenmittel mindern. Was dazu führt, dass weniger Risiko transferiert werden kann und somit in höheren Kapitalanforderungen für den Versicherer resultiert.

Das Kapital, welches für das Gegenparteiausfallrisiko gehalten werden muss, hängt von der Bonität des Rückversicherers ab. Allgemein gilt, je geringer das Bonitätsrating des Rückversicherers, desto höher sind die Kapitalanforderungen für den Versicherer unter Solvency II.

Ob es sinnvoll ist das Langlebigkeitsrisiko mittels Rückversicherung zu hedgen, hängt von der Höhe der Rückversicherungsprämie und der Höhe der Kapitalanforderung für das Gegenparteiausfallrisiko ab. Falls beide zu hoch sind, macht es trotz reduzierter Solvenzkapitalanforderung natürlich keinen Sinn.

Die Rückversicherungsprämie und das Gegenparteiausfallrisiko hängen von 2 wesentlichen Faktoren ab, nämlich von der Größe der rückversicherten Versicherungsbestände und der Bonität des Rückversicherers.

- Falls größere Versicherungsbestände rückversichert werden, kann der Rückversicherer niedrigere Prämien anbieten. Der Nachteil dabei ist, dass damit das Risiko für den Rückversicherer wächst, was zu einer höheren Kapitalanforderung für das Gegenparteiausfallrisiko führt.

- Ein niedrigeres Bonitätsrating ermöglicht es dem Rückversicherer niedrigere Prämien anzubieten, da dieser weniger Kapital in der Bilanz halten muss. Eine niedrige Bonität führt für den Versicherer zu einer höheren Kapitalanforderung für das Gegenpartei-ausfallrisiko.

Der Versicherer hat also die Wahl zwischen einer hohen Prämie und einer geringen Kapitalanforderung oder umgekehrt. Welche dieser Optionen nun die Bessere ist hängt vom Ziel der Kapitalmanagementstrategie des Versicherers ab.

2.3.2 Rückversicherungsvereinbarungen

Welche möglichen Rückversicherungsvereinbarungen es für das Langlebigkeitsrisiko gibt, wird in [P⁺02] behandelt. Hier geht Pitacco speziell auf die Lebensrenten ein. Auf die einzelnen Rückversicherungsarten wird im folgenden Kapitel unter den Abschnitten 3.3 und 3.4 genauer eingegangen.

Wie bereits erwähnt, kann das Langlebigkeitsrisiko unter anderem mittels verschiedener Rückversicherungsvereinbarungen gehedgt werden. Diese wären:

1. Die *Summenexzedentenrückversicherung*, welche auf den abtretenden Teil der hohen Rentenbeträge zielt.
2. Die *Schadensexzedentenrückversicherung (XL-Vertrag)*, bei der der Rückversicherer den letzten Teil der Rente übernimmt, sobald eine gewisse Dauer überschritten wird (zum Beispiel, falls das Alter von 85 Jahren überschritten wird, siehe Abbildung 2.2).
3. Die *Stop-Loss Rückversicherung*, welche darauf abzielt die erforderliche Portfolioreserve zu decken (wie in Abbildung 2.3 veranschaulicht).
4. Die *Quotenrückversicherung* wäre laut [CFvBM20] auch eine Möglichkeit. Hier teilen Versicherer und Rückversicherer zu einem fixierten Prozentsatz die Prämien und Verluste. Dies ermöglicht es dem Versicherer einen Teil des Risikos und der Prämie zu behalten und den Rest dem Rückversicherer zu übertragen.

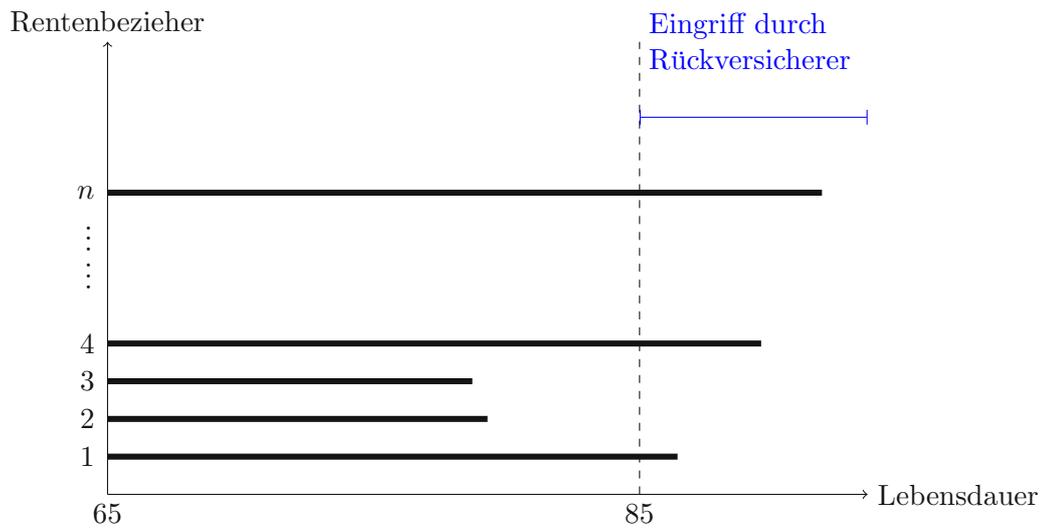


Abbildung 2.2: Eine XL-Rückversicherungsvereinbarung

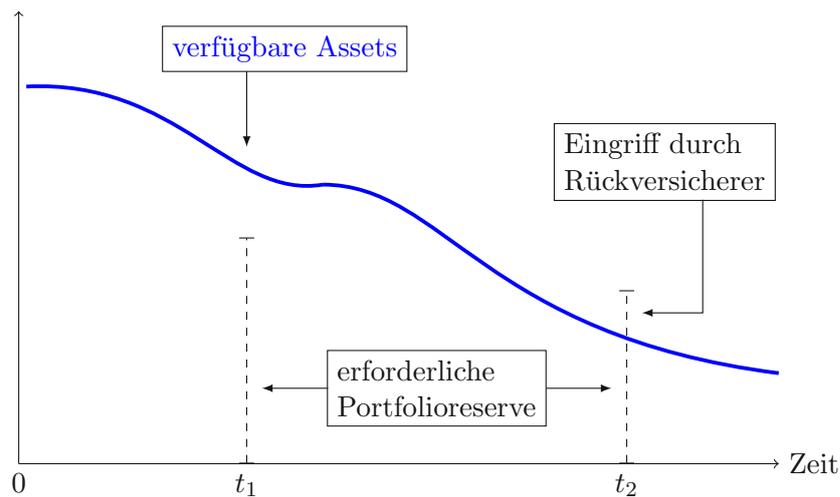


Abbildung 2.3: Eine Stop-Loss Rückversicherungsvereinbarung

Rückversicherungsvereinbarungen über einen kurzen bis mittellangen Zeitraum sind von Vorteil, da man hohe Sicherheitszuschläge auf die Rückversicherungsprämie vermeidet. Aus diesem Grund können Stop-Loss Verträge eine interessante Rückversicherungsdeckung darstellen.

Im Falle einer Stop- Loss Vereinbarung versucht man durch den Eingriff eines Rückversicherers den Zedenten vor der Insolvenz zu bewahren, die durch systematische als auch zufällige Mortalitätsabweichungen entstehen würde. Den Effekt solcher Abweichungen kann man dadurch erkennen, wenn man die verfügbaren Assets des Versicherers zu einem Zeitpunkt mit seinen erforderlichen Portfolioreserven vergleicht (siehe Abbildung 2.3). Die Portfo-

lioreserve wird dazu benötigt, um den Verpflichtungen gegenüber der Rentenempfänger, die bis zu diesem Zeitpunkt überlebt haben, nachzukommen. Daher kann der Eingriff des Rückversicherers also von diesem Vergleich abhängig sein.

Ein zu kurzer Zeitraum kann die zufälligen Abweichungen verstärken, während ein zu langer Zeitraum ein hohes Langlebigkeitsrisiko für den Rückversicherer impliziert und sich dadurch eine sehr hohe Rückversicherungsprämie, aufgrund des hohen Sicherheitszuschlages, ergibt.

3 Rückversicherung

In diesem Kapitel wird die Rückversicherung allgemein erklärt und einige Begriffe dazu definiert. Danach wird auf die verschiedenen Rückversicherungsarten und entsprechenden Prämienberechnungen genauer eingegangen. Das gesamte Kapitel beruht auf [Car83] und [Pfe86].

3.1 Allgemeines und Definitionen

Die Abgrenzung der Rückversicherung und einige allgemeine Definitionen werden genauer aus [Pfe86, Kapitel A. Grundsätzliches II. Begriffsbestimmung der Rückversicherung] und [Car83, Chapter 1: The role and development of reinsurance, Chapter 13: Life reinsurance] entnommen.

Allgemein ist die Rückversicherung (*Zession*) eine Form der Versicherung. Daher sind viele Prinzipien und Anwendungen des Versicherungsgeschäftes genauso für die Rückversicherung zutreffend. Sie ist sogar ein selbstständiger Versicherungsweig.

Ungewisse zukünftige Ereignisse betreffen den Rückversicherer, wie auch die meisten Versicherer, da diese zu Verlusten führen können. Die Rückversicherung bietet somit einen Schutz für den Erstversicherer (*Zedenten*) gegen Verluste, die aufgrund von Leistungen, die er an seine Versicherungsnehmer im Falle eines Versicherungseintritts zahlen müsste, entstehen würden. Für den Erstversicherer ergibt sich dadurch die Möglichkeit Fälle zu versichern, die er aufgrund der Höhe und Gefährlichkeit normalerweise nicht versichern würde. So kann mit Hilfe eines Rückversicherers der notwendige Ausgleich des Geschäftes des Erstversicherers hergestellt werden.

Eine der wichtigsten Aufgaben eines Versicherungsunternehmens ist es Gefahren rechtzeitig zu erkennen und sich dann mittels einer passenden Rückversicherungsform zu schützen.

„Die Rückversicherung ist die Versicherung der von dem Versicherer übernommenen Gefahr“, so lautet die Legaldefinition in § 779 Abs. 1 HGB. [Pfe86, S.12]

Ein *Versicherungsvertrag* ist ein Vertrag zwischen zwei Parteien, wobei eine Partei der Versicherer ist und die andere Partei ein Mitglied der Öffentlichkeit (z.B. eine natürliche Person, ein Unternehmen, eine Behörde oder eine wohltätige Einrichtung). Der Versicherer verspricht seinem Vertragspartner bei Eintritt eines ungewissen aber spezifizierten Ereignisses zu entschädigen oder ihm eine gewisse Summe auszuzahlen und im Gegenzug verspricht der Versicherte eine Prämie an den Versicherer zu zahlen.

Ein *Rückversicherungsvertrag* unterscheidet sich durch folgende 3 Punkte von anderen Versicherungsverträgen:

1. Ein Rückversicherungsvertrag wird zwischen zwei Versicherern abgeschlossen, wobei ein Versicherer die Rolle des Rückversicherers (*Zessionär*) übernimmt und der andere die Rolle des Rückversicherten (*Zedent*). Hier bestehen also keinerlei Rechtsbeziehungen zwischen dem Rückversicherer und dem Versicherungsnehmer.
Ein Rückversicherer kann wiederum einen Rückversicherungsvertrag mit einem anderen Versicherer abschließen, hier wird dann von *Retrozession* gesprochen. Wobei man hier den Rückversicherer, der sein Risiko abgibt, *Retrozedent* nennt und der Rückversicherer, der das Risiko versichert, ist der *Retrozessionär*.
2. Den wesentlichen Gegenstand des Rückversicherungsvertrags bildet die vom Erstversicherer übertragene Gefahr, das sogenannte *Originalrisiko*. Dieses Risiko kann wiederum noch andere Gefahrenerelemente enthalten, wie zum Beispiel Währungs- oder Transferrisiken. Das Originalrisiko kann Eigentum, eine Person oder eine Leistung, die Schäden oder Verlusten ausgesetzt ist, sein.
Daher sichert sich ein Versicherer direkt gegen Ereignisse, die zu wirtschaftlichen Verlusten führen könnten, ab. Für den Rückversicherer ist dieses Primärisiko nur zu dem Teil interessant, den er aufgrund des Rückversicherungsvertrages zu tragen hat.
3. Alle Rückversicherungsverträge, Lebensrückversicherungsverträge mit eingeschlossen, unterliegen dem *Schadenersatzprinzip*. Es gibt jedoch Versicherungsverträge, die dieses Prinzip nicht erfüllen. Das Schadenersatzprinzip besagt, dass der Versicherte keine Gewinne aus dem Eintritt des Schadenfalls ziehen darf. Er soll die resultierende Schadenhöhe möglichst genau ersetzt bekommen. In der Praxis leisten Rückversicherer nur einen Teil der Schadenzahlung, den anderen Teil hat der Erstversicherer selbst zu tragen. [Koc18]

Nach [ZG08] versucht der Erstversicherer mittels einer Rückversicherungsvereinbarung folgende Vorteile zu gewinnen:

- Reduktion der Schadenzahlungen
- Reduktion der Schwankung
- Erhöhung der Sicherheit im Bestand

Der Erstversicherer gibt im Gegenzug einen Teil seines Geschäftes an den Rückversicherer ab.

Nachfrage für Lebensrückversicherung

Die Lebensversicherung unterscheidet sich von der Nicht-Lebensversicherung in den folgenden Punkten:

1. Lebensversicherungsverträge haben meist eine lange Laufzeit und sind vom Versicherer nicht bzw. nur sehr schwer kündbar.
2. Die Verwendung der jährlichen Prämien in der Lebensversicherung, welche:
 - das Mortalitätsrisiko und die Verwaltungskosten gleichmäßig über die Laufzeit verteilen und
 - Reserven über die Laufzeit aufbauen, um die endgültigen Forderungen zu decken.
3. Die variierenden Absicherungs- und Einsparmöglichkeiten für verschiedene Arten von Lebensversicherungsverträgen. Es gibt viele Möglichkeiten Lebensversicherungsverträge zu gestalten.

Der primäre Grund, warum man einen Lebensrückversicherungsvertrag abschließt, ist die Absicherung gegen das Mortalitätsrisiko. Lebensversicherer haben den Vorteil, dass ihnen Sterbetafeln zur Verfügung stehen, mit denen sie sehr gute Vorhersagen über die Mortalität ihrer Versicherungsnehmer aus einem Portfolio in jedem Jahr treffen können. Nichtsdestotrotz kann es vorkommen, dass die vorhergesagte Anzahl der Tode von der tatsächlichen Anzahl signifikant abweicht.

Der Hauptgrund für die Lebensrückversicherung ist, dass der Erstversicherer Schwankungen ausgesetzt ist, die aufgrund einer geringen Anzahl an Versicherten im Gesamtportfolio entstehen können. Vor allem neue Versicherungsunternehmen leiden darunter, da sie zu Beginn meistens nur einen kleinen Bestand aufweisen. Aber auch große und gut etablierte Versicherer können diesen Schwankungen ausgesetzt sein, nämlich in speziellen Versicherungszweigen, in denen der Bestand noch recht klein ist.

Nachdem nun die Abgrenzung vom Rückversicherungsvertrag zu anderen Versicherungsverträgen und die Rolle der Rückversicherung in der Lebensversicherung behandelt wurden, werden im nächsten Abschnitt die juristischen Formen der Rückversicherung erklärt.

3.2 Formen der Rückversicherung

Dieser Abschnitt basiert größtenteils auf [Pfe86, Kapitel B. Formen der Rückversicherung] und ebenso teilweise auf [Car83, Chapter 6: Facultative Reinsurance].

Zunächst gab es jahrhundertlang nur die fakultative, also die freiwillige, Rückdeckung von Einzelrisiken. Erst im 19. Jahrhundert hat die Entwicklung der Rückversicherung ihren Lauf richtig aufgenommen, sodass neue Rückversicherungsformen entwickelt wurden. Deshalb findet man in der heutigen Praxis des Rückversicherers neben der fakultativen Rückversicherung zusätzlich die obligatorische, also die verpflichtende, Rückversicherung.

Bevor die Einteilung in die proportionale und nicht-proportionale Rückversicherung besprochen wird, werden zunächst die Begriffe fakultative Rückversicherung und obligatorische Rückversicherung erläutert.

3.2.1 Die fakultative Rückversicherung

Normalerweise läuft der Abschluss von Versicherungsverträgen so ab, dass beide Vertragsparteien den Inhalt von Leistung und Gegenleistung möglichst genau im Vorhinein darstellen und danach die Bedingungen festlegen.

Bei der *fakultativen Rückversicherung* hat der Erstversicherer die freie Wahl eine Rückversicherungsvereinbarung anzubieten und der Rückversicherer hat dann die Möglichkeit die angebotene Vereinbarung anzunehmen oder abzulehnen. Der Rückversicherer hat somit vor Eingehen der Verpflichtung die Möglichkeit sich ein genaueres Bild der einzelnen Risiken zu machen, die er übernehmen soll.

Die Bedeutung der fakultativen Rückversicherung ist im Vergleich zur obligatorischen aufgrund der kostspieligen Rückversicherungsabgaben, bei denen die Höhe und die Einwilligung der Rückversicherung bis zur Annahmeerklärung ungewiss ist, zurückgegangen. In den letzten Jahren ist allerdings die Bedeutung der fakultativen Rückversicherung wieder gestiegen, vor allem in der Feuer- und Haftpflichtversicherung, in der Lebens-, Transport- und Luftfahrtversicherung sowie in den technischen Versicherungszweigen. Das kann man darauf zurückführen, dass die anfallenden erhöhten Kosten für die Rückversicherer nur eine niedrigere Rückversicherungsprovision erlauben.

Die **Verwendung fakultativer Rückversicherung** kann unter den folgenden Umständen praktisch sein:

1. Wenn die benötigte Rückversicherung, die auf Vertragsbasis angeboten wird, für den Rückversicherer nicht attraktiv ist.
Dies wäre beispielsweise der Fall, wenn ein Unternehmen nur wenig Rückversicherung pro Jahr für einen speziellen Versicherungszweig benötigt. Denn dann könnten die potenziellen Verluste der rückversicherten Risiken im Vergleich zu den jährlichen Rückversicherungsprämien so hoch sein, dass ein Vertrag instabil wäre. Instabil im dem Sinne, dass man jedes Jahr den äußerst variablen Gesamtschadenskosten ausgesetzt ist.
2. Wenn
 - a) das rückzuversichernde Risiko nicht in die bestehenden Verträge eines Versicherungsunternehmens fallen würde. Falls es zum Beispiel einer ausgeschlossenen Risikoklasse angehört.
 - b) die Versicherungssumme Vertragsgrenzen überschreitet.
 - c) das Risiko von solcher Art ist, dass der Versicherer es nicht an seinen Vertrag abtreten will, aufgrund einer potenziellen destabilisierenden Wirkung.

Der Versicherer strebt daher eine fakultative Rückversicherung besonders gegen Spitzenrisiken und gefährliche Risiken an.

3. Wenn ein Versicherungsunternehmen seine Aufnahmekapazität erhöhen will.

In der **Lebensversicherung** wird auf rein fakultativer oder auf fakultativ-obligatorischer Basis rückversichert, da sie gelegentlich eher lose Verbindungen bevorzugen. Unter fakultativ-obligatorischer Rückversicherung (*Open Cover*) versteht man ein Abkommen, welches es dem Erstversicherer ermöglicht genau zu bezeichnende Risikokategorien rückzuversichern und den Rückversicherer zur Übernahme dieser Risiken verpflichtet. Somit fällt bei solch einer Form die Möglichkeit des Rückversicherers das Risiko abzulehnen weg. Der Rückversicherer erhält dadurch nicht nur einzelne Risiken, sondern einen breiteren Umfang und kann dadurch eine höhere Rückversicherungsprovision vergüten.

In einigen Ländern hat die fakultative Rückversicherung eine besondere Bedeutung für die Versicherung von erhöhten Risiken. Dabei handelt es sich um Personen, die eine erhöhte Todes- oder Invaliditätswahrscheinlichkeit aufweisen. Falls der Rückversicherer einwilligt das erhöhte Risiko rückzuversichern, dann erhält der Erstversicherer die Bedingungen dafür. Bei diesen Bedingungen kann es sich um Staffelungen der Versicherungssumme, Prämienzuschläge oder Ausschluss gewisser Gefahren handeln. Falls die Versicherung zustande kommt, hat sich der Rückversicherer dazu verpflichtet die Risiken zum Teil oder vollständig zu übernehmen.

3.2.2 Die obligatorische Rückversicherung

Diese Form von Rückversicherung findet man in der Praxis in der Ausprägung der Quoten- und Summenexzedentenrückversicherung sowie in der Schadenrückversicherung mit den Hauptformen der Schadenexzedenten- und Jahresüberschadenrückversicherung (*Stop-Loss Rückversicherung*).

„Die obligatorische Rückversicherung bindet die Parteien wesentlich stärker aneinander als vereinzelte fakultative Rückversicherungsabgaben.“ [Pfe86, S.27]

Beim *obligatorischen Rückversicherungsvertrag* ist der Erstversicherer dazu verpflichtet einen bestimmten Umfang an Risiken rückzuversichern und der Rückversicherer dazu verpflichtet diese Risiken zu übernehmen. Da der Rückversicherer nicht jedes einzelne Risiko beurteilen kann, muss der Zedent die Prämien angemessen festsetzen und die Schäden so regeln, wie er es im gemeinsamen Interesse für richtig hält. Der Erstversicherer hat die Pflicht bei allen Entscheidungen die Interessen des Rückversicherers sowie seine eigenen zu wahren. Verletzt er diese, so wird der Rückversicherer von der Pflicht, die Risiken zu übernehmen, befreit. In der Praxis kann sich der Zedent stets darauf verlassen, dass der Rückversicherer seiner Leistungspflicht nachkommt, solange er selbst gesunde Geschäftsgrundsätze anwendet.

Bei der obligatorischen Rückversicherung ist daher das *Prinzip von Treu und Glauben* besonders wichtig, da der Rückversicherer keine genaueren Informationen zu den Einzelrisiken hat. Dieses Prinzip besagt, dass Versicherer und Rückversicherer einander vertrauen und sich nicht gegenseitig kontrollieren. [Koc18]

Nun werden die technischen Formen der Rückversicherung besprochen, nämlich die Aufteilung in proportionale und die nicht-proportionale Rückversicherung.

3.3 Proportionale Rückversicherung

Die Definition und Funktion der proportionalen Rückversicherung wird aus [Pfe86, Kapitel D. Proportionale Rückversicherung] und [Car83, Chapter 3: Principles and practice of reinsurance; Chapter 5: Forms of reinsurance] entnommen.

Eine *proportionale Rückversicherung* liegt vor, wenn das rückzuversichernde Risiko zu einem festen Prozentsatz zwischen Zedent und Rückversicherer aufgeteilt wird. Somit übernimmt der Rückversicherer die Teil- oder Totalschäden zu diesem Prozentsatz und erhält wiederum den prozentualen Anteil der Originalprämie. Die proportionale Rückversicherung kommt in zwei Hauptformen vor, nämlich der Summenexzedentenrückversicherung und der Quotenrückversicherung.

Bei der Summenexzedentenrückversicherung ergibt sich der prozentuale Anteil des Rückversicherers aus dem Verhältnis der Versicherungssumme zum Selbstbehalt. Wenn beispielsweise der Selbstbehalt des Zedenten 100.000,00 EUR und die Versicherungssumme 200.000,00 EUR betragen, dann erhält der Rückversicherer 50 % der Prämien und beteiligt sich zu 50 % an den Total- und Teilschäden (siehe Abbildung 3.1).

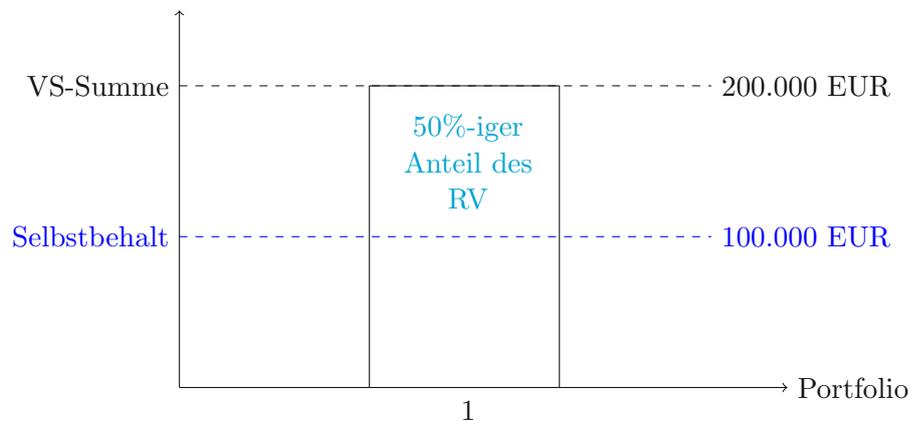


Abbildung 3.1: (Summen-)Exzedentenrückversicherung

Im Gegensatz dazu wird bei der Quotenrückversicherung der prozentuale Anteil des Rückversicherers ohne Rücksicht auf die Höhe der Versicherungssummen der einzelnen Verträge festgelegt. Daher hat ein Quotenrückversicherungsvertrag nicht diese ausgleichende Wirkung, die der Summenexzedentenvertrag besitzt. Aus diesem Grund werden oft zusätzliche Rückversicherungsmaßnahmen benötigt.

Abbildung 3.2 gibt einen guten Überblick über die Formen der proportionalen Rückversicherung.

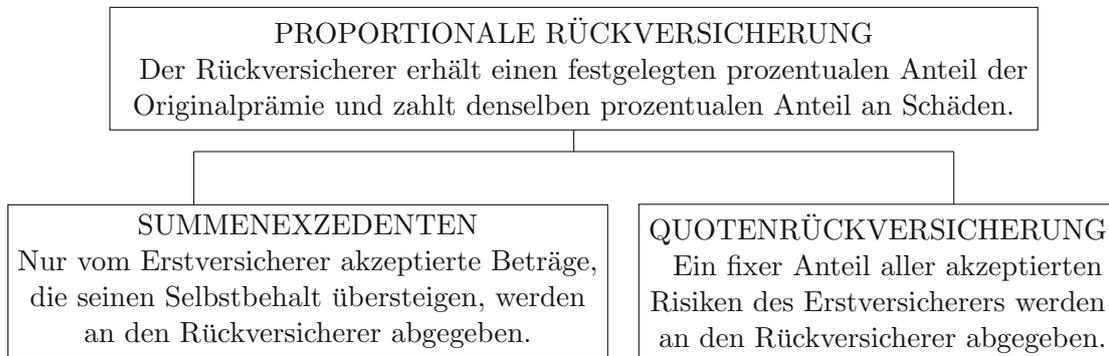


Abbildung 3.2: Formen der proportionalen Rückversicherung

3.3.1 Summenexzedentenrückversicherung

Die (*Summen-*)*Exzedentenrückversicherung* zählt sowohl zur ältesten als auch wichtigsten Form der proportionalen Rückversicherung. Wie bei der Quotenrückversicherung übernimmt der Rückversicherer einen gewissen Anteil des Risikos. Dabei erhält er wiederum denselben Anteil der Prämie und übernimmt zum selben Anteil die Schäden. Der Unterschied zur Quotenrückversicherung ist, dass der Rückversicherer bei der Summenexzedentenrückversicherung das Risiko zu dem Anteil übernimmt, zu dem der Selbstbehalt überstiegen wird.

Die Summenexzedentenrückversicherung wird vor allem in den Versicherungszweigen angewandt, in denen die Versicherungssummen der gedeckten Risiken stark voneinander abweichen. Sie ist besonders dafür geeignet den notwendigen Ausgleich im Bestand des Zedenten herzustellen, welcher durch überaus hohe Schäden gefährdet werden kann. Am häufigsten findet man sie in der Feuer-, Unfall- und Lebensversicherung.

Falls der Zedent für eine Exzedentenrückversicherung entscheidet, ist die erste und wichtigste Aufgabe einen Selbstbehalt (*Maximum*) festzusetzen. Dafür muss man zunächst *Risikoklassen* bilden, für die jeweils unterschiedliche Selbstbehalte festzulegen sind. Generell gilt, je höher der Gefährdungsgrad, desto niedriger wird der Selbstbehalt bemessen.

In den meisten Fällen wird der Selbstbehalt nach der Versicherungssumme festgesetzt. Eine andere Möglichkeit ist es den Selbstbehalt nach dem wahrscheinlichen Höchstschaden (engl: *Probable Maximum Loss = PML*) zu bestimmen. Es besteht allerdings die Gefahr, dass es zu einer starken Fehleinschätzung des PML durch den Zedenten kommen kann. Dagegen kann sich der Rückversicherer schützen, indem er für extreme Fälle (z.B. bei einer Verschätzung um mehr als 50 %) eine Beteiligung des Zedenten am Schadenteil des Rückversicherers vereinbart.

Für normale Risiken wird der Selbstbehalt in der **Lebensrückversicherung** als fester Betrag festgelegt. Es ist üblich diesen Betrag ab einer gewissen Altersgrenze zu reduzieren. Die Höhe des Selbstbehaltes wird in Abhängigkeit von den besonderen Gegebenheiten der Lebensversicherung festgelegt. Wenn es um die Versicherung von erhöhten Risiken in der Lebensversicherung geht, ist bei der Einschätzung das *Irrtumsrisiko* zu berücksichtigen. Deshalb stuft man hier oft Selbstbehalte nach der Schwere der übernommenen Risiken ab.

Den Teil der Versicherung, der den Selbstbehalt übersteigt, nennt man *Exzedent*. Die Höhe des Exzedenten wird beschränkt, damit der Rückversicherer die Höhe seiner Verpflichtung kennt. Dessen Höhe wird so beschränkt, dass er nur eine begrenzte Anzahl von Vielfachen des Maximums umfasst, z.B. 10 Maxima. Zusätzlich muss aber auch ein vernünftiges Verhältnis zwischen Selbstbehalt und Rückversicherungsabgabe bestehen.

Größere Erstversicherungsgesellschaften schließen Rückversicherungsvereinbarungen mit mehreren Rückversicherern ab und teilen den Exzedenten prozentual unter ihnen auf.

3.3.2 Quotenrückversicherung

Die *Quotenrückversicherung* zählt zu den einfachsten Formen der Rückversicherung. Hier verpflichtet sich der Rückversicherer dazu alle Versicherungen eines bestimmten Versicherungszweiges oder eines Teiles davon zu einem festen Prozentsatz (*Quote*) zu übernehmen. Hier wird die Höhe der einzelnen Versicherungen nicht berücksichtigt, was bei der Exzedentenrückversicherung sehr wohl der Fall ist.

Mathematisch wird die Quotenrückversicherung in [ZG08] folgendermaßen beschrieben: Jeder Einzelschaden X wird in den Anteil des Erstversicherers X_E und den Anteil des Rückversicherers X_R zerlegt:

$$X = X_E + X_R.$$

Die beiden Anteile haben folgende Form:

$$X_E = \alpha X \quad \text{und} \quad X_R = (1 - \alpha)X.$$

Hier ist $0 \leq \alpha \leq 1$ der Selbstbehalt und $(1 - \alpha)$ die Quote.

Die Quotenrückversicherung schützt in einfacher und wirksamer Weise gegen die Kumulierung kleiner und mittlerer Schäden und wird besonders häufig in den folgenden zwei Fällen gewählt. Einerseits, wenn ein neugegründetes Unternehmen seinen Geschäftsbetrieb beginnt und die Erfahrung des Rückversicherers sehr gefragt ist. Und andererseits, wenn neue Versicherungsarten und Gefahren rückversichert werden, da der Zedent damit zunächst sein eigenes Risiko stark begrenzen kann und der Rückversicherer so einen breiteren Ausgleich erzielt.

In der **Lebensrückversicherung** findet die Quotenrückversicherung besonders bei neuen oder expandierenden Gesellschaften Anwendung. Für die Rückversicherung normaler Risi-

ken gilt hier, dass sowohl die Schwankungen im Schadenverlauf gedeckt werden, als auch eine hohe Beteiligung des Rückversicherers an den Abschlusskosten entsteht. Erhöhte Risiken werden meist durch Exzedenten- sowie Quotenrückversicherungsverträge gedeckt.

Sowohl die Bearbeitung als auch die Abwicklung der Quotenrückversicherung ist für beide Vertragspartner einfach und kostensparend, da der Prozentsatz und somit auch die Höhe des Selbstbehaltes von Anfang an festgelegt wird.

3.4 Nicht-proportionale Rückversicherung

Dieser Abschnitt beruht größtenteils auf [Pfe86, Kapitel E. Nicht-proportionale Rückversicherung] und zum Teil auf [Car83, Chapter 3: Principles and practice of reinsurance; Chapter 13: Life reinsurance]. Die mathematische Herangehensweise wird wieder aus [ZG08] entnommen.

Bei der *nicht-proportionalen Rückversicherung* wird die Leistung des Rückversicherers ausschließlich durch die Höhe des Schadens bestimmt. Es findet somit keine proportionale Aufteilung des einzelnen Risikos und der dafür erhobenen Prämie statt. Sie findet in verschiedenen Versicherungszweigen Anwendung, unter anderem als Schutzmaßnahme gegen Kumulierungen und Katastrophenschäden.

Erstversicherer gehen diese Art von Rückversicherungsvereinbarungen seltener ein, da spezielle Sachkunde benötigt wird. Die nicht-proportionale Rückversicherung kann nur erfolgreich betrieben werden, wenn es viele Verträge dieser Art im Bestand gibt. Trotzdem ist die Gefahr hoher Verluste wesentlich höher als bei der Quoten- und Exzedentenrückversicherung.

Andererseits gewinnt die nicht-proportionale Rückversicherung immer mehr an Beliebtheit, da sie das Exposure bezüglich Einzelrisiken effizienter minimiert als die proportionale Rückversicherung und somit für stabile Geschäftsergebnisse des Erstversicherers sorgt. Ein weiterer Vorteil ist, dass man sich mit nicht-proportionaler Rückversicherung administrativen Aufwand und somit auch Kosten spart. Der ständige Anstieg in administrativen Kosten führt zu einem Umschwung von proportionaler zu nicht-proportionaler Rückversicherung, sogar in Versicherungszweigen, die eigentlich auf proportionaler Basis rückversichert haben.

In der nicht-proportionalen Rückversicherung gibt es zwei Hauptformen:

1. die **Schadenexzedentenrückversicherung** (engl: *Excess of Loss*, abgekürzt XL)
2. die **Jahresüberschadenrückversicherung** (engl: *Stop Loss*)

Für beide Formen gilt, dass der Zedent einen Betrag bzw. bei der Stop-Loss Rückversicherung einen Prozentsatz seiner Jahresprämie festsetzt, bis zu dem er den Schaden selbst übernimmt. Diesen Betrag bezeichnet man als *Priorität*. Den darüber hinausgehenden Schaden übernimmt dann der Rückversicherer, allerdings nur bis zu einer betragsmäßig bzw. wiederum als Prozentsatz der Jahresprämie des Zedenten festgesetzten Übernahmemaximums (*Haftstrecke*).

Falls ein besonders hoher Rückversicherungsschutz benötigt wird, gibt es die Möglichkeit mehrere Schadenexzedentenrückversicherungen nacheinander zu schalten. Dabei ergibt sich beispielsweise die Priorität der zweiten Schadenexzedentendeckung aus der Summe der Priorität und der Haftstrecke des ersten Abschnittes usw.

Beispiel:

Priorität des Zedenten	100.000,00 EUR
Haftung des ersten Rückversicherers	400.000,00 EUR
<hr/>	
Priorität des zweiten Schadenexzedenten	500.000,00 EUR
Haftung des zweiten Rückversicherers	1.000.000,00 EUR
<hr/>	
Priorität des dritten Schadenexzedenten	1.500.000,00 EUR
Haftung des dritten Rückversicherers	unbegrenzte Haftung

Ein weiterer Grund für diese Aufteilung in verschiedene Haftungsabschnitte (*Layer*) kann die Trennung in einen unteren Bereich und in einen Katastrophenbereich sein. Der untere Bereich ist jener, bei dem regelmäßig mit Schäden zu rechnen ist (*working cover*) und im Katastrophenbereich befinden sich nur ganz außergewöhnliche Schäden.

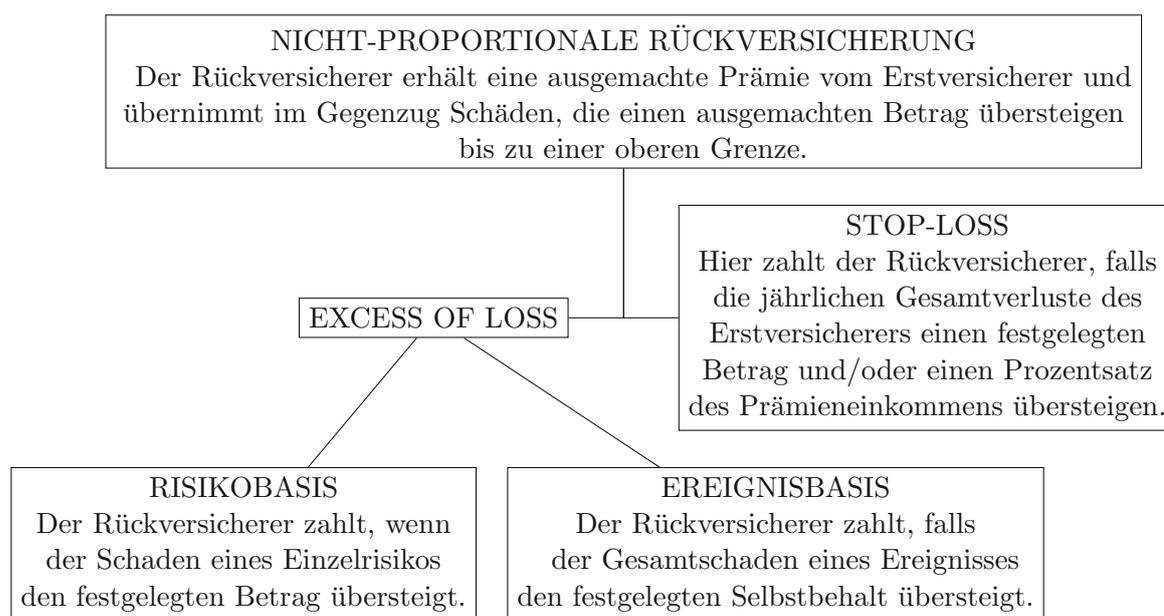


Abbildung 3.3: Formen der nicht-proportionalen Rückversicherung

Abbildung 3.3 gibt einen guten Überblick über die Formen der nicht-proportionalen Rückversicherung. Nun wird auf die einzelnen Formen etwas genauer eingegangen und die mathematische Formulierung jener betrachtet.

3.4.1 Schadenexzedentenrückversicherung

Die *Schadenexzedentenrückversicherung* deckt entweder den Schaden eines **Einzelrisikos** oder sie deckt als **Kumulschadenexzedent** alle Schäden, die auf mehrere Risiken oder Polizzen infolge eines Schadenereignisses anfallen. In beiden Fällen gilt natürlich, dass die Priorität überschritten werden muss. In letzterem Fall gilt außerdem, dass die Priorität so hoch angesetzt werden muss, dass sie von einem Schaden aus einer Polizza nicht erreicht werden kann. Der Kumulschadenexzedent schützt den Erstversicherer vor Kumulierungen und ergänzt zweckmäßig Exzedenten- und Quotenverträge.

Ob eine Deckung mittels Schadenexzedentenrückversicherung überhaupt möglich ist, hängt von den Besonderheiten des Versicherungszweiges, von den Eigenschaften des Versicherungsbestandes, der zu decken ist, und von der passenden Prämienwahl ab. Verhältnismäßig häufig findet man diese Art von Rückversicherung in Versicherungszweigen, bei denen nur kleine bis mittlere Schäden anfallen und Großschäden eine Ausnahme darstellen.

Für die mathematische Beschreibung der XL-Rückversicherung wird von einem Einzelschaden mit Schadenhöhe X ausgegangen. Der Zedent übernimmt den Anteil bis zur Priorität (*retention level*) M , den restlichen Anteil übernimmt der Rückversicherer. Es gilt:

$$X_E = \begin{cases} X, & \text{falls } X \leq M \\ M, & \text{falls } X > M \end{cases} \quad \text{und} \quad X_R = \begin{cases} 0, & \text{falls } X \leq M \\ X - M, & \text{falls } X > M. \end{cases}$$

Dabei ist X_E der Anteil des Erstversicherers und X_R jener Anteil, den der Rückversicherer übernimmt.

Die Formen der nicht-proportionalen Rückversicherung finden hauptsächlich in der Nicht-Lebens(rück-)versicherung Anwendung und haben noch nicht die gleichen Fortschritte in der Lebensrückversicherung gemacht. Das liegt unter anderem an der kurzen Laufzeit der XL-Verträge, was einen Nachteil für den Erstversicherer darstellt. Grundsätzlich werden Schadenexzedentenverträge auf einer jährlichen Basis abgeschlossen, gelegentlich kann die Laufzeit auch zwei bis drei Jahre betragen. Falls sich die Schadenerfahrung schlechter entwickelt als angenommen kann der Rückversicherer die Erneuerung des Vertrages ablehnen oder im besten Fall höhere Prämien verlangen.

In der **Lebensrückversicherung** findet eine Schadenexzedentendeckung hauptsächlich Anwendung bei der Rückversicherung von Kumulrisiken. Hier wird der Fall rückversichert, dass mehrere Personen durch **ein** Unfallereignis oder **eine** Naturkatastrophe ums Leben kommen. In der Rückversicherungsvereinbarung wird die Priorität und die Mindestanzahl der betroffenen Personen festgelegt. Daher steigt auch die Nachfrage für XL-Rückversicherungsverträge, da Versicherer immer mehr diesem Kumulrisiko für Todesfallleistungen ausgesetzt sind.

3.4.2 Stop-Loss Rückversicherung

Die *Stop-Loss Rückversicherung* bietet für den Zedenten einen umfassenden Schutz gegen Schwankungen des Schadenverlaufs eines Versicherungszweiges oder eines Teilbestandes. Der Rückversicherer übernimmt jenen Schaden, der die Priorität übersteigt. Bei dieser Form der Rückversicherung wird die Priorität als Prozentsatz der verdienten oder verrechneten Jahresprämie ausgedrückt. Hier wird nicht berücksichtigt, ob die Priorität aufgrund von hohen Einzelschäden oder kleinen bis mittleren Schäden überschritten wird.

Stop-Loss Verträge kommen prinzipiell selten vor. Die häufigsten Anwendungsfälle sind in der Sturm- und Hagelversicherung. In anderen Versicherungszweigen kommt sie nicht oft zur Anwendung. Bei der Festsetzung der Priorität ist darauf zu achten, dass der Zedent auch im Falle eines ungünstigen Geschäftsverlaufs einen Teil des Schadens selbst zu tragen hat und keinen sicheren Gewinn ermöglicht bekommt.

Bei der mathematischen Formulierung der Stop-Loss Rückversicherung geht man vom Gesamtschaden mit Schadenhöhe S aus. Dieser wird zerlegt in den Anteil des Erstversicherers S_E und den Anteil des Rückversicherers S_R :

$$S = S_E + S_R.$$

Man definiert die Stop-Loss Rückversicherung analog wie die XL-Rückversicherung. Der einzige Unterschied ist, dass sie auf den Gesamtschaden statt auf den Einzelschaden wirkt:

$$S_E = \begin{cases} S, & \text{falls } S \leq d \\ d, & \text{falls } S > d \end{cases} \quad \text{und} \quad S_R = \begin{cases} 0, & \text{falls } S \leq d \\ S - d, & \text{falls } S > d. \end{cases}$$

Hier wird mit d der *stop loss point* bezeichnet. Die Höhe des Gesamtschadens für den Zedenten wird somit genau bei d abgestoppt.

Obwohl die Stop-Loss Rückversicherung für viele Versicherer attraktiv zu sein scheint, ist die Verwendung in der Praxis limitiert, da sie Nachteile mit sich bringen kann. Genauso wie XL-Verträge werden auch Stop-Loss Verträge auf einer jährlichen Basis abgeschlossen. Auf der anderen Seite konnte man sehen, dass je größer das Portfolio ist, desto stabiler ist die Schadenerfahrung. Dies führt wiederum zu einer Reduktion des benötigten Prämienatzes für die Stop-Loss Rückversicherung.

3.4.3 Prämienberechnung

Bei der proportionalen Rückversicherung erhält der Rückversicherer einen Anteil der Originalprämie. Bei der nicht-proportionalen Rückversicherung hingegen wird die Rückversicherungsprämie vom einzelnen Originalrisiko losgelöst und pauschal für das gesamte geschützte Portfolio berechnet.

Die **XL-Prämie** setzt sich folgendermaßen zusammen:

1. Aus der **reinen Risikoprämie** π_R , die benötigt wird, um die erwarteten Schäden aus der Rückversicherung zu bezahlen.
2. Ein **Sicherheitszuschlag** λ ist erforderlich, da der Schadenverlauf von Jahr zu Jahr starken Schwankungen ausgesetzt ist.
3. Auch **externe Kosten** κ_E wie z.B. Steuern müssen berücksichtigt werden.
4. Zuletzt muss man noch die **internen Betriebskosten** κ_I des Rückversicherers einbeziehen.

Somit gilt für die Prämie:

$$\pi = \pi_R + \underbrace{\lambda + \kappa_E + \kappa_I}_L.$$

Die letzten drei Komponenten kann man als Zuschlag L (*engl: Loading*) zusammenfassen. Zu den internen Kosten des Rückversicherers zählen unter anderem Risikomanagementkosten. Dazu gehören auch Kosten, die entstehen, falls der Rückversicherer einen Teil seines Geschäfts in Rückversicherung gibt. [Koc18]

Obige und folgende Grundsätze gelten ebenso für die **Stop-Loss-Prämie**. Der einzige Unterschied ist, dass bei der Ermittlung der Prämienrate für die Stop-Loss-Prämie der Schadensatz aus dem Verhältnis der verdienten Prämie zu den angefallenen Schäden ermittelt wird. Bei Schadenexzedentenverträgen wird statt der verdienten die verrechnete Prämie verwendet. Auch hier hängen die Zuschläge von den zu erwartenden Schwankungen des Schadenverlaufs ab.

Es gibt drei verschiedene Methoden die Risikoprämie π_R zu berechnen. Welches Verfahren schließlich verwendet wird, hängt von bestimmten Umständen ab, wie zum Beispiel der Sparte, der Priorität und der Haftstrecke. Diese drei Verfahren werden nun genauer vorgestellt.

Pay-Back-Verfahren

Beim Pay-Back-Verfahren basiert die Kalkulation der Prämie auf Wiederkehrperioden bestimmter Großschadenereignisse. Man versucht zu ermitteln in welchen Zeitperioden Großschäden zu erwarten sind. Man beschränkt sich hier auf Totalschäden, die die gesamte Haftstrecke verbrauchen. Wenn man beispielsweise davon ausgeht, dass alle fünf Jahre ein Totalschaden zu erwarten ist, dann muss man die Rückversicherungsprämie so ansetzen, dass aus ihr innerhalb von fünf Jahren (Pay-Back-Periode) einmal eine ganze Haftstrecke bezahlt werden kann. Diese Methode lässt sich allerdings nur auf wenige Sparten anwenden und ist eher für einfache Kontrollrechnungen geeignet.

Burning-Cost-Verfahren

Beim Burning-Cost-Verfahren basiert die Kalkulation der Prämie auf der Schadenerfahrung der Vergangenheit. Hier ermittelt man welche Schadenbelastung sich in den letzten Jahren unter Berücksichtigung der zukünftigen Priorität und Haftstrecke für den Vertrag ergeben hätte und diese setzt man ins Verhältnis zur jeweiligen Prämieinnahme.

$$BC_t = \frac{\text{Schadenanteil des Rückversicherers in } t}{\text{Prämieinnahme des Zedenten in } t},$$

wobei BC_t die *Burning Cost* eines vergangenen Schadenjahres t ist. Die *mittlere Burning Cost* erhält man, indem man die Anteile des Rückversicherers aller vergangenen Schadenjahre durch die Prämieinnahmen des Zedenten aller vergangenen Schadenjahre dividiert:

$$\text{mittlere Burning Cost} = \frac{\sum_t \text{Schadenanteile des Rückversicherers in } t}{\sum_t \text{Prämieinnahmen des Erstversicherers in } t}.$$

Die mittlere Burning Cost ist somit jener Prozentsatz, der den Anteil an der Prämieinnahme des Zedenten bestimmt, den der Rückversicherer im aktuellen Jahr erhält. Dabei sind jedoch einige Punkte zu beachten:

1. Aufgrund der anhaltenden Inflation und der ständigen Erhöhung der Löhne sind die Statistiken nicht mehr repräsentativ, da vergangene Schäden mit geringeren Aufwendungen zu regulieren waren als zukünftig zu erwartende.
2. Zum Zeitpunkt der Prämienkalkulation gilt in der Regel, dass viele Schäden noch nicht vollständig reguliert wurden. Das endgültige Schadensmaß ist somit zu dem Zeitpunkt noch nicht klar. Die Berücksichtigung von Spätschäden ist allerdings noch schwieriger. Das sind Schäden, die bereits eingetreten sind, aber noch nicht als solche erkannt worden sind bzw. nicht gemeldet wurden. Die dafür erforderliche Rückstellung nennt man Spätschadenreserve (engl: IBNR = incurred but not reported).
3. Da der Versicherungsbestand des Erstversicherers bei dieser Methode aus vergangenen Daten besteht, muss geprüft werden, ob dieser immer noch repräsentativ für sein Portfolio ist.

Das Burning-Cost-Verfahren eignet sich für die Prämienkalkulation von Schadenexzedenten mit niedrigen Prioritäten, bei der eine hinreichende Schadenerfahrung vorhanden ist.

Exposure-Verfahren

Beim Exposure-Verfahren basiert die Kalkulation der Prämie auf der Risikostruktur des gedeckten Portfolios. Hier teilt der Zedent sein Portfolio in möglichst homogene Teilbestände und innerhalb dieser wird nochmals nach Versicherungs- und Deckungssumme gegliedert. Diese Daten werden dann dem Rückversicherer zur Verfügung gestellt und anhand dieser

Aufteilung wird die Rückversicherungsprämie ermittelt. Dieses Verfahren wird im Falle hoher Prioritäten angewandt.

Da es oft zu Schwankungen in den Beständen des Versicherers kommen kann, gibt es auch die Möglichkeit eine vom Schadensatz abhängige **variable Prämie** zu wählen. Dabei vereinbart man eine *Höchst-* und eine *Mindestprämie*. Je nachdem wie sich die Schäden entwickeln, kann sich innerhalb dieser Grenzen die Prämie um einen Zuschlag erhöhen. Dieser ist meist ein Faktor oder ein Prozentsatz der Basisprämie.

In [Car83, Chapter 13: Life reinsurance] wird eine weitere Methode vorgestellt die Rückversicherungsprämie zu berechnen, nämlich die Risikoprämienmethode (engl: *risk premium method*).

Risikoprämienmethode

Die Risikoprämienmethode ist so konzipiert, dass sie Abhilfe für die sogenannte *Death Strain* schafft. Die Definition wird aus [Koc18] entnommen. Die *Death Strain* ist die Differenz zwischen der Leistung S , die der Versicherer tatsächlich zu zahlen hat, und dem zum Zeitpunkt $t + 1$ vorhandenen Deckungskapital ${}_{t+1}V$.

$$\text{DeathStrain} = S - {}_{t+1}V.$$

Die *Death Strain* ist daher jene Summe, die der Versicherer zum Zeitpunkt $t + 1$ zahlen muss, falls der Versicherungsnehmer im Intervall $[t, t + 1]$ stirbt.

Die Grundidee der Risikoprämienmethode ist, dass der Betrag, der jedes Jahr rückversichert wird, gleich dem Anteil der originalen Versicherungssumme, der den Selbstbehalt übersteigt, abzüglich der dafür angelegten Reserven ist. Der Erstversicherer baut jedes Jahr Reserven auf, welche einzelnen Polizzen zugeschrieben werden. Dadurch vermindert sich die Risikosumme und folglich auch die Rückversicherung.

Bei dieser Methode muss die Rückversicherungsprämie jedes Jahr neu berechnet werden, um die Minderung des rückversicherten Risikos widerzuspiegeln.

Zu Beginn jedes Versicherungsjahres erhält man die Risikoprämie, indem man die Risikoprämienrate für das erlangte Lebensalter auf die Risikosumme für dieses Jahr anwendet.

Die *Risikoprämienrate* P_x für eine x -jährige Person ist folgendermaßen definiert:

$$P_x = v^{1/2}q_x,$$

wobei

- $v = \frac{1}{1+i}$ der Diskontierungsfaktor mit Zinsrate i ist und

- q_x die Wahrscheinlichkeit eines x -Jährigen vor Erreichen des Alters $x + 1$ zu sterben ist.

Die *Rückversicherungsrisikoprämie* $\pi_{x,t}^R$ für einen x -Jährigen im Jahr t ist das Produkt der Risikoprämienrate einer x -jährigen Person und der Risikosumme R im Jahr t :

$$\pi_{x,t}^R = P_x R_t.$$

4 Deep Learning

In diesem Kapitel wird zuerst auf die Grundlagen des Machine Learnings eingegangen, da diese die Basis des Deep Learnings bilden. Danach wird mithilfe der Definition eines Perzeptrons der Aufbau eines Neuronalen Netzwerks erklärt. Weiters wird darauf eingegangen, wie solch ein Netzwerk letztendlich trainiert bzw. optimiert wird. Hierfür werden einige Hyperparameter und Optimierungsalgorithmen wie Backpropagation, (stochastisches) Gradienten Abstiegsverfahren und Algorithmen mit adaptiver Lernrate vorgestellt.

4.1 Grundlagen Machine Learning

Die Grundlagen des Machine Learnings werden aus [GBC16, Chapter 5: Machine Learning Basics] entnommen.

Deep Learning ist eine spezielle Art des Machine Learnings. Aus diesem Grund ist es wichtig die grundsätzlichen Prinzipien des Machine Learnings nachvollziehen zu können, um Deep Learning zu verstehen.

4.1.1 Lernende Algorithmen

Ein Machine Learning Algorithmus ist ein Algorithmus, der imstande ist von Daten zu lernen. Was genau mit Lernen gemeint ist, beschreibt [Mit97] folgendermaßen:

„Ein Computerprogramm lernt von der Erfahrung E in Bezug auf eine Klasse von Aufgaben (engl: *tasks*) T und einem Performancemaß P , falls sich die Performance der Aufgaben in T , welche mittels P gemessen wird, mit der Erfahrung E verbessert.“

Die Aufgabe T

Machine Learning ermöglicht es uns Aufgaben zu lösen, die für herkömmliche Algorithmen zu aufwändig wären. Machine Learning Aufgaben werden üblicherweise in dem Sinne beschrieben, wie das System ein *Beispiel* verarbeiten soll. Dabei ist ein Beispiel eine Sammlung von *Eigenschaften bzw. Merkmalen*, die von einem Objekt oder Ereignis gemessen wurden, welches das Machine Learning System bearbeiten soll. Manchmal werden Beispiele auch als *Datenpunkte* bezeichnet. Eine Sammlung von Beispielen bzw. Datenpunkten ist ein *Datensatz*. Der Prozess des Lernens selbst ist nicht die Aufgabe. Ein Beispiel ist typischerweise ein Vektor $\mathbf{x} \in \mathbb{R}^n$, wobei jeder Vektoreintrag x_i ein Merkmal ist. Beispielsweise sind die Merkmale eines Bildes die Werte der Pixel des Bildes.

Das Performancemaß P

Um einen Machine Learning Algorithmus bewerten zu können, benötigt man ein quantitatives Maß der Performance. Dieses Maß ist üblicherweise von der Aufgabe T abhängig. Für eine Aufgabe wie die Klassifikation wird die *Accuracy* (deutsch: *Genauigkeit*) des Modells gemessen. Bei der Klassifikation bestimmt das Programm, zu welchen k Kategorien der Input gehört. Die Accuracy ist der Anteil an Beispielen, für welche das Modell den richtigen Output erzeugt. Man erhält eine äquivalente Information, indem man die *Fehlerrate* misst, also den Anteil an Beispielen, für den das Modell einen falschen Output liefert.

Für eine Aufgabe wie die Dichteschätzung macht es keinen Sinn die Accuracy oder die Fehlerrate zu messen. Hier muss der Algorithmus die Struktur der Daten lernen und wissen, wo sich die Beispiele clustern und wo sie unwahrscheinlich auftreten. Die häufigste Herangehensweise ist es die durchschnittliche Log-Wahrscheinlichkeit anzugeben, die das Modell den Beispielen zuordnet.

Meist ist man daran interessiert wie gut ein Machine Learning Algorithmus mit Daten abschneidet, die er zuvor noch nicht gesehen hat. Daher wird ein sogenannter *Testdatensatz* herangezogen, welcher sich von den Daten unterscheidet, mit denen das Machine Learning System trainiert wurde. Das Performancemaß zeigt dann wiederum wie gut das System abschneidet. Die richtige Wahl des Performancemaßes ist nicht eindeutig und oft schwierig.

Die Erfahrung E

Machine Learning Algorithmen können in *supervised* (deutsch: *überwacht*) und *unsupervised* (deutsch: *unüberwacht*) eingeteilt werden, je nachdem welche Art von Erfahrung sie während des Lernprozesses haben.

Unsupervised Learning Algorithmen bearbeiten bzw. „erfahren“ einen Datensatz, welcher viele Merkmale besitzt, und lernt nützliche Eigenschaften der Struktur des Datensatzes. Im Falle des Deep Learnings will man die gesamte Wahrscheinlichkeitsverteilung eines Datensatzes lernen, wie zum Beispiel bei der Dichteschätzung. Ein anderes Beispiel für unsupervised Algorithmen wäre Clustering, welches den Datensatz in Cluster ähnlicher Beispiele einteilt.

Unsupervised Algorithmen betrachten daher mehrere Beispiele von Zufallsvektoren \mathbf{x} und versuchen die Wahrscheinlichkeitsverteilung $p(\mathbf{x})$ oder interessante Eigenschaften der Verteilung zu lernen.

Supervised Learning Algorithmen bearbeiten bzw. „erfahren“ einen Datensatz mit Merkmalen, wobei jedem Datenpunkt ein *Label* oder ein *Target* zugeordnet wird. Ein typisches Beispiel für supervised Algorithmen wäre die Klassifikation, wobei jeder der k Kategorien ein Label zugeschrieben wird.

Supervised Algorithmen betrachten mehrere Beispiele eines Zufallsvektors \mathbf{x} und dem dazugehörigen Wert oder Vektor \mathbf{y} und lernen dann \mathbf{y} aus \mathbf{x} vorherzusagen, meistens mittels Schätzung der bedingten Wahrscheinlichkeit $p(\mathbf{y}|\mathbf{x})$.

4.1.2 Kapazität, Overfitting und Underfitting

Die Herausforderung des Machine Learnings ist, dass der Algorithmus auf neuen und noch nicht zuvor gesehenen Inputs gut abschneidet. Falls dies der Fall ist, dann bezeichnet man diese Fähigkeit als *Generalisierung* (engl: *generalization*).

Ein Machine Learning Modell wird trainiert, indem man einen Trainingsdatensatz heranzieht und ein Fehlermaß auf diesem Datensatz berechnet, den sogenannten *Trainingsfehler*. Diesen kann man verkleinern, indem man die Parameter entsprechend anpasst. Somit handelt es sich um ein Optimierungsproblem. Was allerdings Machine Learning von einer Optimierung unterscheidet ist, dass man den *Generalisierungsfehler*, welcher auch als *Testfehler* bezeichnet wird, möglichst gering halten will. Der Generalisierungsfehler ist der Erwartungswert des Fehlers eines neuen Inputs. Den Generalisierungsfehler eines Machine Learning Modells erhält man typischerweise, indem die Performance des Testdatensatzes gemessen wird.

Wie gut ein Machine Learning Algorithmus abschneidet hängt von folgenden zwei Fähigkeiten ab:

1. Den Trainingsfehler klein zu halten.
2. Den Unterschied zwischen Trainings- und Testfehler klein zu halten.

Diese beiden Faktoren entsprechen den zwei zentralen Herausforderungen in Machine Learning, nämlich dem **Underfitting** und dem **Overfitting**. *Underfitting* liegt vor, wenn es dem Modell nicht möglich ist einen hinreichend kleinen Trainingsfehler zu erhalten. *Overfitting* bedeutet, dass der Unterschied zwischen Trainings- und Testfehler zu groß ist.

Es ist jedoch möglich das Modell so zu beeinflussen, dass es eher over- oder underfittet, indem man seine **Kapazität** verändert. Die *Kapazität* eines Modells ist die Fähigkeit eine Vielzahl an Funktionen zu fitten. Modelle mit niedriger Kapazität haben Schwierigkeiten den Trainingsdatensatz zu fitten. Modelle mit hoher Kapazität können den Datensatz overfitten. Eine Möglichkeit die Kapazität zu verändern ist es einen *Hypothesenraum* zu wählen. Das ist jene Menge an Funktionen, die der Algorithmus als Lösung auswählen kann. Machine Learning Algorithmen schneiden am besten ab, wenn die Kapazität zur Komplexität der Aufgabe und zur Anzahl der Trainingsdaten passt.

Regularisierung

Die Regularisierung (engl: *regularization*) ist jede beliebige Abänderung des Algorithmus, um den Generalisierungsfehler, aber nicht den Trainingsfehler zu verringern. Prinzipiell gilt, dass es keinen besten Machine Learning Algorithmus gibt und auch keine beste Form der Regularisierung. Die Wahl ist abhängig von der speziellen Aufgabe, die zu lösen ist.

4.1.3 Hyperparameter und Validierungsdatensatz

In Machine Learning gibt es zwei Arten von Parametern. Einerseits gibt es Modellparameter und andererseits Parameter, die man anpasst, um das Modell besser und schneller zu gestalten. Letztere Art der Parameter werden *Hyperparameter* genannt. Diese kontrollieren die Optimierungsfunktionen und die Modellauswahl während der Trainingsphase des Lernalgorithmus. Die Wahl der Hyperparameter konzentriert sich darauf sicherzustellen, dass das Modell den Trainingsdatensatz während der Lernphase weder unter- noch overfittet. [PG17, Chapter 1: A Review of Machine Learning].

Ein anderes Beispiel für Hyperparameter sind jene, die die Modellkapazität kontrollieren. Solche Hyperparameter wählen immer die größtmögliche Kapazität. Dies würde jedoch in overfitting resultieren. Um dieses Problem zu lösen kommt ein sogenannter *Validierungsdatensatz* (engl: *validation set*) ins Spiel. Für die Konstruktion des Validierungsdatensatzes werden Trainingsdaten herangezogen. Diese werden in zwei disjunkte Datensätze geteilt. Einer dieser Teildatensätze wird verwendet, um die Parameter zu lernen. Der andere ist der Validierungsdatensatz und wird verwendet, um den Generalisierungsfehler während oder nach der Trainingsphase zu bestimmen und die Hyperparameter entsprechend anzupassen. Üblicherweise werden ca. 80 % der Trainingsdaten für das Trainieren und 20 % für die Validierung verwendet.

4.1.4 Motivation für Deep Learning

Machine Learning Algorithmen sind in der Lage eine Vielzahl an wichtigen Problemstellungen zu lösen. Allerdings stellt sich heraus, dass herkömmliche Machine Learning Algorithmen besonders zur Lösung von Problemen betreffend der Sprach- oder Objekterkennung unzureichend sind. Das Scheitern herkömmlicher Algorithmen hat unter anderem die Entwicklung des Deep Learnings motiviert.

Deep Learning liefert ein leistungsstarkes Framework für supervised Learning. Ein Neuronales Netz ermöglicht es sehr komplexe Funktionen darzustellen, indem man mehrere Layer und mehrere Einheiten innerhalb eines Layers hinzufügt. Aufgaben, die darin bestehen, einen Inputvektor auf einen Outputvektor abzubilden, können mittels Deep Learning ausgeführt werden. Es müssen jedoch ausreichend große Modelle und ausreichend große Datensätze mit gelabelten Trainingsbeispielen gegeben sein. [GBC16, Part II: Modern Practical Deep Networks]

4.2 Aufbau Neuronaler Netzwerke

Folgender Abschnitt wird aus [A⁺18, Chapter 1: Introduction to Neural Networks], [PG17, Chapter 2: Foundation of Neural Networks and Deep Learning] und [GBC16, Chapter 6: Deep Feedforward Networks] entnommen.

Die Idee Neuronaler Netze ist von der Funktionsweise des menschlichen Gehirns inspiriert. Das menschliche Nervensystem enthält Zellen, sogenannte *Neuronen*. Die einzelnen

Neuronen sind über *Axonen* und *Dendriten* miteinander verbunden. Die Verbindung zwischen Axonen und Dendriten stellt die Synapsen dar. Neuronen kommunizieren miteinander, indem Impulse über die Synapsen weitergeleitet werden, die so stark sind, dass dabei Chemikalien freigesetzt werden. Die Stärke der synaptischen Verbindungen ändert sich oft aufgrund externer Impulse. Die Veränderung beschreibt den Lernvorgang lebender Organismen. Diesen biologischen Mechanismus will man mittels künstlicher Neuronaler Netze simulieren.

Der Aufbau Neuronaler Netzwerke kann folgendermaßen definiert werden:

- Anzahl der Neuronen
- Anzahl der Layer
- Art der Verbindung zwischen den Layern

Im Folgenden werden Neuronale Netzwerke mit nur einem Layer (single-Layer) und mit mehreren Layern (multi-Layer) betrachtet. Im single-Layer Netzwerk wird eine Menge an Inputs mittels einer linearen Funktion direkt auf den Output abgebildet. Diese einfache Form eines Neuronalen Netzes wird auch *Perzeptron* (engl: *perceptron*) genannt. Ein multi-Layer Netzwerk besteht aus einem Inputlayer, einem oder mehreren hidden Layern und einem Outputlayer. Diese Art von Netzwerken bezeichnet man auch als *Feedforward Neuronale Netzwerke*.

4.2.1 Das Perzeptron

Das **Perzeptron** ist die einfachste Form eines Neuronalen Netzwerks. Es ist ein lineares Modell, welches für binäre Klassifizierung verwendet wird. Das Perzeptron enthält einen einzigen Inputlayer und einen Outputknoten. Der Aufbau des Perzeptrons wird in Abbildung 4.1 dargestellt.

Angenommen jedes Trainingsbeispiel hat die Form (\bar{X}, y) , wobei $\bar{X} = [x_1, \dots, x_d]$ d -viele Merkmalvariablen sind und $y \in \{-1, 1\}$ die Labels darstellen. Die Labels sind in diesem Fall *beobachtete Werte* der binären Klassenvariable, die im Rahmen der Trainingsdaten gegeben sind. Ziel ist es, die Klassenvariablen für unbeobachtete Werte vorherzusagen.

Der Inputlayer enthält d -viele Knoten, welche die d Merkmale $\bar{X} = [x_1, \dots, x_d]$ mit Gewichten $\bar{W} = [w_1, \dots, w_d]$ auf einen Outputknoten übertragen. Der Inputlayer für sich genommen führt keine Berechnungen durch. Die Berechnung der linearen Funktion $\bar{W} \cdot \bar{X} = \sum_{i=1}^d w_i x_i$ findet im Outputknoten statt. Anschließend wird das Vorzeichen dieses Wertes verwendet, um \hat{y} aus $[x_1, \dots, x_d]$ vorherzusagen:

$$\hat{y} = \text{sign}\{\bar{W} \cdot \bar{X}\} = \text{sign}\left\{\sum_{i=1}^d w_i x_i\right\}.$$

Die Signumfunktion bildet den Wert auf -1 oder $+1$ ab, was wiederum einer binären Klassifikation entspricht. Die Signumfunktion dient in diesem Setting als eine *Aktivierungsfunktion*. In Machine Learning werden verschiedenste Aktivierungsfunktionen verwendet, um verschiedene Arten von Modellen zu simulieren. Genauer zu diesem Thema wird später in Abschnitt 4.3 behandelt.

Da der Inputlayer keinerlei Berechnungen durchführt, sondern nur die Merkmalvariablen überträgt, besitzt das Perzeptron nur einen einzigen *rechnerischen* (engl: *computational*) Layer und wird deshalb als single-Layer Netzwerk betrachtet.

Manchmal gibt es Settings, in denen die Vorhersage verzerrt (engl: *biased*) ist. Das wäre beispielsweise der Fall, wenn die Merkmalvariablen der binären Klassifizierung von $\{-1, +1\}$ um den Mittelwert zentriert wären und dieser jedoch nicht 0 ist.

In solch einem Fall ist die obige Herangehensweise nicht hinreichend und es muss eine zusätzliche Biasvariable b miteinbezogen werden:

$$\hat{y} = \text{sign}\{\bar{W} \cdot \bar{X} + b\} = \text{sign}\left\{\sum_{i=1}^d w_i x_i + b\right\}.$$

Der Bias kann als Gewicht einer Kante eines sogenannten *Biasneurons* hinzugefügt werden. Hierfür wird ein Neuron verwendet, welches immer den Wert 1 an den Outputknoten überträgt. Das Gewicht der Kante, welche das Biasneuron und den Outputknoten verbindet, liefert die Biasvariable.

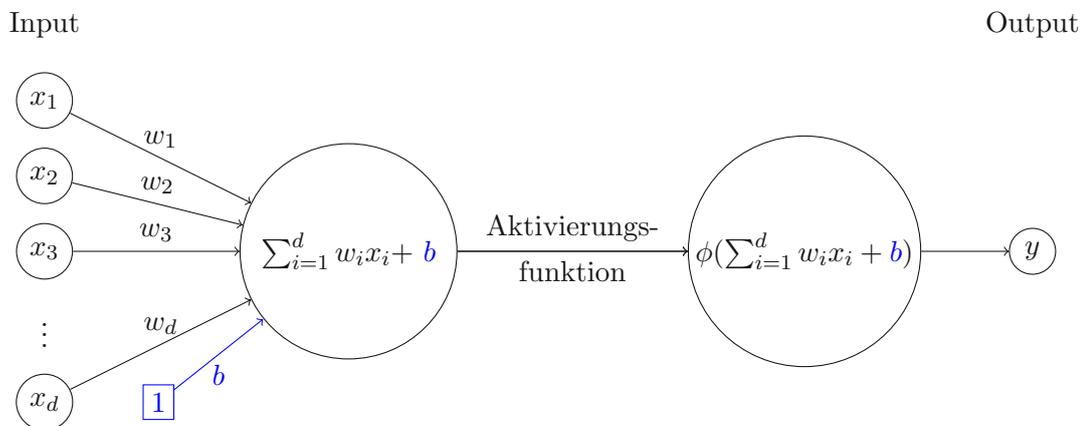


Abbildung 4.1: Aufbau eines Perzeptrons mit Aktivierungsfunktion ϕ und Bias b

4.2.2 Feedforward Neuronale Netzwerke

Feedforward Neuronale Netzwerke oder auch **Multilayer Perceptrons** (MLPs) stellen die grundlegenden Deep Learning Modelle dar. Das Ziel ist es eine Funktion f^* zu approximieren, wobei $y = f^*(\mathbf{x})$. Dabei wird eine Abbildung $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta})$ definiert und das

Netzwerk lernt die Parameterwerte θ so, dass die Funktion bestmöglich approximiert wird.

Multilayer Neuronale Netzwerke besitzen einen Inputlayer, einen oder mehrere hidden Layer und einen Outputlayer. Jeder Layer enthält ein oder mehrere Neuronen. Jedes einzelne Neuron entspricht dem oben erwähnten Perzeptron, wobei die Aktivierungsfunktion je nach Nutzen des Layers im Netzwerk unterschiedlich gewählt wird. Jeder Knoten in einem Layer ist mit jedem Knoten im nächsten Layer verbunden. Im Gegensatz zum Perzeptron enthält ein Multilayer Neuronales Netzwerk mehrere rechnerische Layer. Die Architektur des Feedforward Neuronalen Netzwerkes wird in Abbildung 4.2 veranschaulicht.

In [BGTW19] findet sich folgende Definition für Feedforward Neuronale Netzwerke.

Definition 4.2.1. Seien $L, N_0, N_1, \dots, N_L \in \mathbb{N}$, $\phi : \mathbb{R} \rightarrow \mathbb{R}$ und für jedes $l \in \{1, \dots, L\}$ sei $S_l : \mathbb{R}^{N_{l-1}} \rightarrow \mathbb{R}^{N_l}$ eine affine Funktion. Dann nennt man eine Funktion $F : \mathbb{R}^{N_0} \rightarrow \mathbb{R}^{N_L}$ definiert durch

$$F(x) = S_L \circ F_{L-1} \circ \dots \circ F_1 \quad \text{mit} \quad F_l = \phi \circ S_l \quad \text{für} \quad l = 1, \dots, L - 1,$$

(Feedforward) Neuronales Netzwerk.

Die Aktivierungsfunktion ϕ wird komponentenweise angewandt.

L bezeichnet die Anzahl der Layer, N_1, \dots, N_{L-1} sind die Dimensionen der hidden Layer und N_0, N_L die Dimensionen der Input- und Outputlayer.

Für jedes $l \in \{1, \dots, L\}$ ist die affine Funktion S_l gegeben durch $S_l(x) = A^l x + b^l$ für $A^l \in \mathbb{R}^{N_l \times N_{l-1}}$ und $b^l \in \mathbb{R}^{N_l}$.

Für jedes $i \in \{1, \dots, N_l\}$ und $j \in \{1, \dots, N_{l-1}\}$ wird die Zahl A_{ij}^l interpretiert als Gewicht jener Kante, welche den i -ten Knoten des $l-1$ -ten Layers und den j -ten Knoten des l -ten Layers verbindet. Die Anzahl der Gewichte des Netzwerkes, die nicht Null sind, ist die Summe der Einträge der Matrix A^l und der Vektoren b^l , die nicht Null sind, jeweils mit $l = 1, \dots, L$

[BGTW19] erläutert weiters, dass Neuronale Netzwerke multivariate Funktionen beliebig gut approximieren können. Hierzu wird definiert, dass $\mathcal{NN}_{\infty, d_0, d_1}^\phi$ die Menge der Neuronalen Netzwerke mit Aktivierungsfunktion ϕ ist, die von $\mathbb{R}^{d_0} \rightarrow \mathbb{R}^{d_1}$ abbilden.

Theorem 4.2.1 (Universal Approximation, [Hor91]). Angenommen ϕ ist beschränkt und nicht konstant. Dann gelten folgende Aussagen:

- Für jedes endliche Maß μ auf $(\mathbb{R}^{d_0}, \mathcal{B}(\mathbb{R}^{d_0}))$ und $1 \leq p < \infty$ ist die Menge $\mathcal{NN}_{\infty, d_0, 1}^\phi$ dicht in $L^p(\mathbb{R}^{d_0}, \mu)$.
- Falls zusätzlich gilt, dass $\phi \in C(\mathbb{R})$, dann ist $\mathcal{NN}_{\infty, d_0, 1}^\phi$ dicht in $C(\mathbb{R}^{d_0})$ bezüglich der Topologie der gleichmäßigen Konvergenz auf kompakten Mengen.

Da jede Komponente des \mathbb{R}^{d_1} -wertigen Neuronalen Netzwerkes ein \mathbb{R} -wertiges Neuronales Netz ist, lässt sich obiges Resultat auch auf Neuronale Netzwerke $\mathcal{NN}_{\infty, d_0, d_1}^\phi$ mit $d_1 > 1$ anwenden.

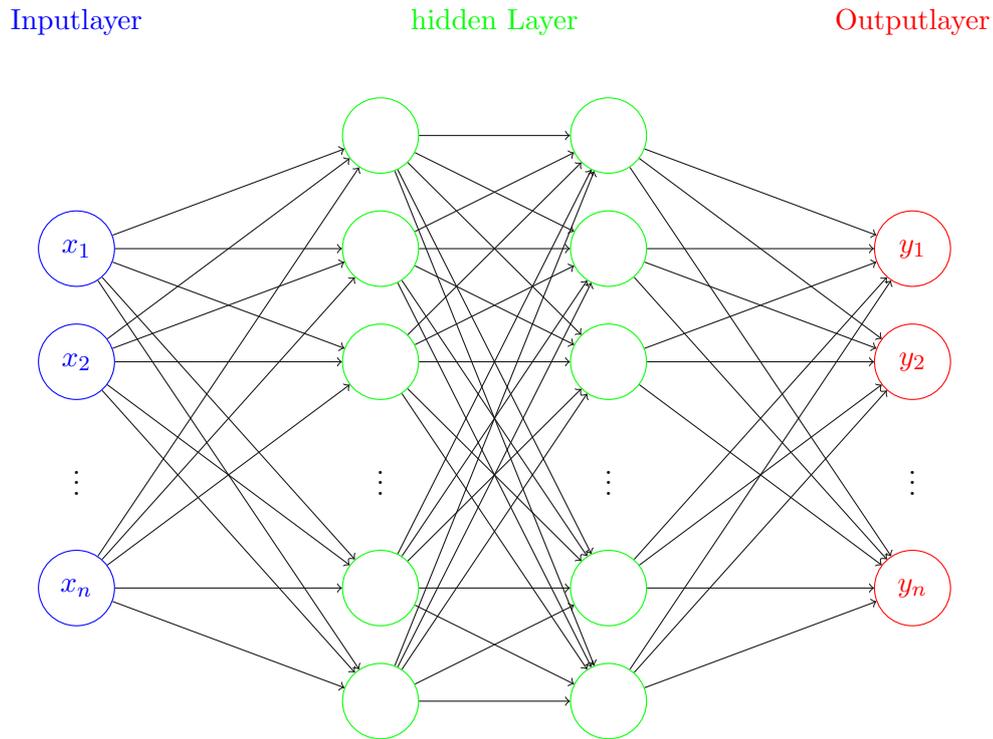


Abbildung 4.2: Feedforward Neuronales Netzwerk mit zwei hidden Layern

4.3 Aktivierungsfunktionen

In diesem Abschnitt werden Aktivierungsfunktionen definiert und erklärt welchen Nutzen sie haben. Die Informationen hierzu werden aus [PG17, Chapter 2: Foundations of Neural Networks and Deep Learning], [A⁺18, Chapter 1: Introduction to Neural Networks] und [Led21] entnommen.

Aktivierungsfunktionen bestimmen das Verhalten eines Neurons. Sie transformieren die Linearkombination bestehend aus Input, Gewichte und Bias und das Ergebnis der Transformation dient als Input für die Knoten im nächsten Layer. Viele nichtlineare Transformationen transformieren die Daten auf einen geeigneten Wertebereich, zum Beispiel auf $(0, 1)$ oder $(-1, 1)$. Wenn ein Neuron einen Wert, der nicht Null ist, einem anderen Neuron übergibt, dann nennt man dieses *aktiviert*.

Die Wahl der Aktivierungsfunktion ist entscheidend für die Erstellung eines Neuronales Netzwerks. Im Falle des Perzeptrons, bei dem es sich um ein binäres Klassifizierungsproblem handelt, wählt man die Signumfunktion als Aktivierungsfunktion. Die Wahl ist jedoch davon abhängig, was vorhergesagt werden soll. Falls die Variable, die man vorhersagen möchte, reellwertig ist, dann ist die Identität als Aktivierungsfunktion eine passende Wahl und liefert dasselbe wie eine Least-Squares Regression.

Identität oder lineare Aktivierung

$$\phi(x) = x \in (-\infty, +\infty).$$

Diese Aktivierungsfunktion wird im Inputlayer des Neuronalen Netzwerks verwendet, da hier die Information unverändert weitergegeben wird.

Die Bedeutung für *nichtlineare Aktivierungsfunktionen* steigt, wenn man vom Perzeptron zu multi-Layer Netzwerken übergeht. Im Folgenden werden einige nichtlineare Aktivierungsfunktionen vorgestellt.

Signum

$$\phi(x) = \begin{cases} -1 & , x < 0 \\ 0 & , x = 0 \\ +1 & , x > 0. \end{cases}$$

Wie in Abschnitt 4.2.1 bereits erwähnt, wird die Signumfunktion im Perzeptron für die binäre Klassifizierung angewandt.

Sigmoid

$$\phi(x) = \frac{1}{1 + e^{-x}} \in (0, 1).$$

Sigmoidfunktionen können, wie alle logistischen Transformationen, Extremwerte und Ausreißer in den Daten reduzieren, ohne sie dabei zu entfernen. Falls die vorherzusagenden Variablen Wahrscheinlichkeiten sein sollen, ist Sigmoid als Aktivierungsfunktion eine passende Wahl.

Tangens hyperbolicus (Tanh)

$$\phi(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \in (-1, 1).$$

Tanh hat eine ähnliche Form wie Sigmoid, allerdings wird Tanh vorgezogen, falls der Output sowohl positive als auch negative Werte annehmen soll. Weiters ist Tanh um den Mittelwert zentriert und ein größerer Gradient erleichtert das Trainieren.

Softsign

$$\phi(x) = \frac{x}{1 + |x|} \in (-1, 1).$$

Genauso wie Tanh ist auch die Softsignfunktion um den Mittelwert zentriert. Softsign hat gegenüber Tanh Vorteile bezüglich des rechnerischen Aufwandes, da Softsign prinzipiell einfacher ist und keine Exponentialterme enthält.

In den letzten Jahren haben *stückweise lineare Aktivierungsfunktionen* immer mehr an Bedeutung gewonnen:

Rectified Linear Unit (ReLU)

$$\phi(x) = \max\{x, 0\} = \begin{cases} 0 & , x < 0 \\ x & , x \geq 0 \end{cases} \in [0, \infty).$$

Die ReLU-Funktion aktiviert einen Knoten erst, wenn der Input größer Null ist. Die Funktion ist eine sehr beliebte Wahl, da sie im Trainingsprozess oft besser abschneidet als die Sigmoidfunktion.

Exponential Linear Unit (ELU)

$$\phi(x) = \begin{cases} a(e^x - 1) & , x < 0 \\ x & , x \geq 0 \end{cases} \in (-a, \infty) \text{ mit } a \in [0, \infty).$$

ELU ist eine Variante von ReLU und verwendet eine logarithmische Kurve, um negative Werte zu definieren. [SSA17]

Hard Tanh

$$\phi(x) = \max\{\min\{x, 1\}, -1\} \in [-1, 1].$$

Der Hard Tanh ist eine robustere Aktivierungsfunktion und sorgt für limitierte Entscheidungsgrenzen.

Softplus

$$\phi(x) = \ln(1 + e^x) \in [0, \infty).$$

Die Softplusfunktion ist die „smoother“ Version der ReLU-Funktion.

Abbildung 4.3 zeigt eine graphische Zusammenfassung der verschiedenen Aktivierungsfunktionen. Es gibt eine Vielzahl weiterer Aktivierungsfunktionen, in dieser Arbeit wird es jedoch bei den oben genannten belassen.

4.4 Verlustfunktionen

Dieser Abschnitt beruht auf [PG17, Chapter 2: Foundation of Neural Networks and Deep Learning].

Verlustfunktionen (engl: *loss functions*) messen, wie gut die Vorhersagen des Neuronalen Netzes in der Trainingsphase abschneiden. Die Idee ist, dass jeder vorhergesagte Output mit dem tatsächlichen Ergebnis verglichen wird und die Metrik dieses Fehlers berechnet wird. Danach werden alle Fehler aufsummiert und der durchschnittliche Fehler berechnet. Der daraus resultierende Wert gibt an, wie gut oder schlecht das Neuronale Netzwerk abschneidet.

Um ein möglichst ideales Ergebnis zu erzielen muss man die Gewichte und Biases so bestimmen, dass der „Verlust“, der durch Fehler in der Vorhersage entsteht, minimal wird. Damit

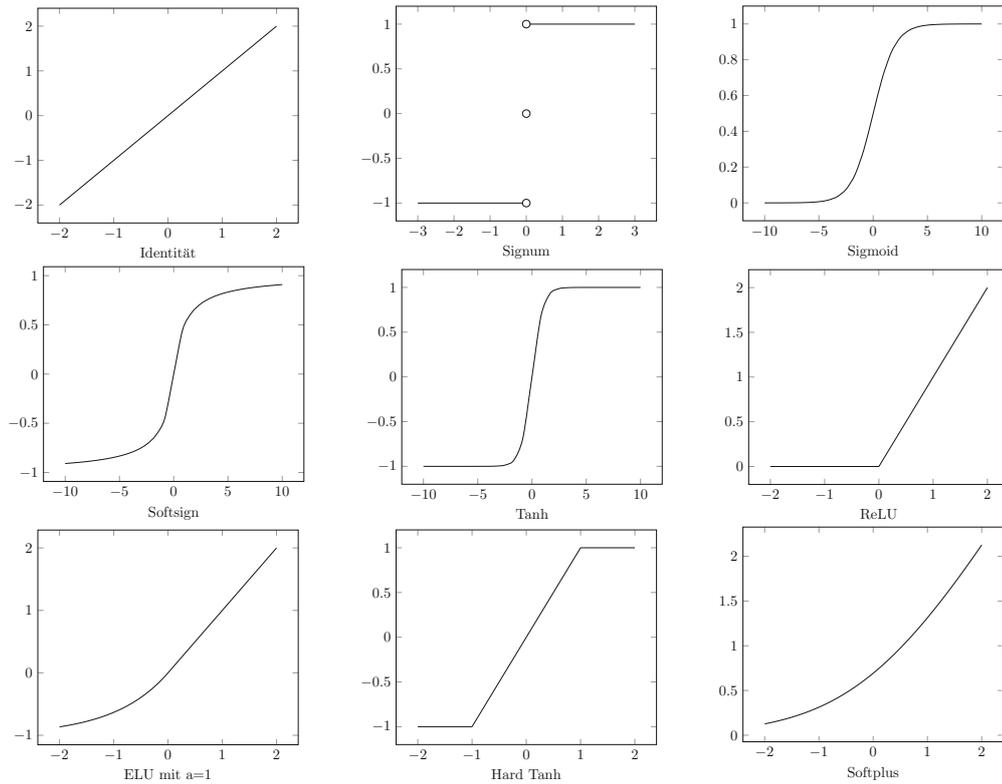


Abbildung 4.3: Aktivierungsfunktionen

wird das Trainieren von Neuronale Netzen zu einem Optimierungsproblem. Meistens können diese Parameter nicht analytisch bestimmt werden. Dann werden sie mittels iterativen Optimierungsalgorithmen, wie beispielsweise dem Gradienten Abstiegsverfahren, approximiert.

Die Wahl der Verlustfunktion hängt davon ab, wie die Outputknoten definiert werden beziehungsweise was die Aufgabe des Neuronale Netzes ist.

Eine häufige Wahl für ein Fehlermaß in einem Regressionsmodell ist der **mittlere quadratische Fehler** (engl: mean squared error) (MSE). Der Output ist reellwertig und der Fehler in der Vorhersage \hat{y} wird quadriert und über alle Datenpunkte gemittelt:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2.$$

Eine ebenso häufige Wahl für eine Verlustfunktion ist der **mittlere absolute Fehler** (engl: mean absolute error) (MAE). Dieser mittelt den absoluten Fehler über den gesamten Datensatz:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|.$$

4.5 Optimierung Neuronaler Netzwerke

In diesem Abschnitt werden verschiedene (iterative) Optimierungsalgorithmen vorgestellt. Informationen hierfür werden aus [PG17, Chapter 2: Foundation of Neural Networks and Deep Learning] und [A⁺18, Chapter 1.3: Training a Neural Network with Backpropagation] herangezogen.

Der Trainingsprozess eines Neuronalen Netzwerks entspricht einem Optimierungsproblem und besteht darin die Gewichte und Biases neu anzupassen, sprich zu erhöhen oder zu verringern, sodass manchen Informationen mehr und anderen weniger Bedeutung zugeschrieben wird. So lernt das Modell welche Merkmale an welchen Output gebunden sind und passt die Gewichte und Biases dementsprechend an.

4.5.1 Backpropagation

Der Backpropagationalgorithmus sorgt dafür, dass der Fehler in einem Deep Learning Modell reduziert wird. Er besteht aus einer Vorwärts- und einer Rückwärtsphase.

- *Vorwärtsphase:* In dieser Phase wird der Input für einen Trainingsvorgang dem Neuronalen Netzwerk übergeben. Dies führt zu Berechnungen in den verschiedenen Layern mit den momentanen Gewichten. Der daraus resultierende, vorhergesagte Output wird dann mit dem tatsächlichen Output verglichen und der Verlust berechnet. Um die Gewichte in jedem Layer upzudaten, wird die Ableitung der Verlustfunktion herangezogen. Dies geschieht in der Rückwärtsphase.
- *Rückwärtsphase:* Im Outputknoten startend, werden die Gradienten der Verlustfunktion in Bezug auf die verschiedenen Gewichte mittels Kettenregel berechnet. Die Berechnung läuft dann rückwärts weiter und die resultierenden Gradienten werden verwendet, um die Gewichte upzudaten. Danach startet der Algorithmus wieder von vorne.

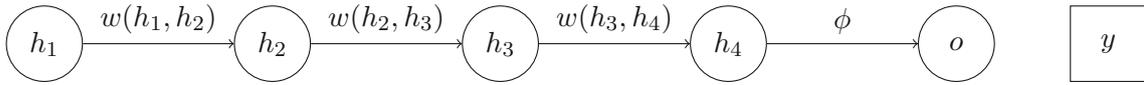
Um zu sehen wie der Algorithmus funktioniert, betrachtet man eine Folge von Knoten h_1, h_2, \dots, h_k gefolgt von einem Output o , mittels welchen die Verlustfunktion L berechnet wird. Weiters wird das Gewicht, welches die Knoten h_r und h_{r+1} verbindet, mit $w(h_r, h_{r+1})$ bezeichnet.

In dem Fall, dass ein einziger Weg von h_1 nach o existiert, kann der Gradient der Verlustfunktion in Bezug auf jedes dieser Gewichte mittels Kettenregel berechnet werden:

$$\frac{\partial L}{\partial w(h_{r-1}, h_r)} = \frac{\partial L}{\partial o} \cdot \left[\frac{\partial o}{\partial h_k} \prod_{i=r}^{k-1} \frac{\partial h_{i+1}}{\partial h_i} \right] \frac{\partial h_r}{\partial w(h_{r-1}, h_r)} \quad \forall r \in \{1, \dots, k\}. \quad (4.1)$$

Um die Formel nachzuvollziehen, wird auf ein einfaches Beispiel eingegangen.

Angenommen es sind vier Knoten h_1, \dots, h_4 gegeben und es führt nur ein Weg von h_1 nach o :



Der Wert y stellt das tatsächliche Ergebnis dar, somit ergibt sich für den Fehler beziehungsweise den Verlust L_0 :

$$\begin{aligned}
 L_0 &= (o - y)^2 \\
 &\downarrow \\
 o &= \phi(h_4) \\
 &\downarrow \\
 h_4 &= w(h_3, h_4) \cdot h_3 + b(h_3, h_4) \\
 &\downarrow \\
 h_3 &= w(h_2, h_3) \cdot h_2 + b(h_2, h_3) \\
 &\downarrow \\
 h_2 &= w(h_1, h_2) \cdot h_1 + b(h_1, h_2).
 \end{aligned}$$

Nun kann man mittels Kettenregel von unten startend nach oben die Ableitungen berechnen:

$$\begin{aligned}
 \frac{\partial L_0}{\partial w(h_1, h_2)} &= \frac{\partial h_2}{\partial w(h_1, h_2)} \cdot \left[\frac{\partial h_3}{\partial h_2} \cdot \frac{\partial h_4}{\partial h_3} \cdot \frac{\partial o}{\partial h_4} \right] \cdot \frac{\partial L_0}{\partial o} = \frac{\partial L_0}{\partial o} \left[\frac{\partial o}{\partial h_4} \cdot \frac{\partial h_3}{\partial h_2} \cdot \frac{\partial h_4}{\partial h_3} \right] \cdot \frac{\partial h_2}{\partial w(h_1, h_2)} \\
 \frac{\partial L_0}{\partial w(h_2, h_3)} &= \frac{\partial h_3}{\partial w(h_2, h_3)} \cdot \left[\frac{\partial h_4}{\partial h_3} \cdot \frac{\partial o}{\partial h_4} \right] \cdot \frac{\partial L_0}{\partial o} = \frac{\partial L_0}{\partial o} \cdot \left[\frac{\partial o}{\partial h_4} \cdot \frac{\partial h_4}{\partial h_3} \right] \cdot \frac{\partial h_3}{\partial w(h_2, h_3)} \\
 \frac{\partial L_0}{\partial w(h_3, h_4)} &= \frac{\partial h_4}{\partial w(h_3, h_4)} \cdot \left[\frac{\partial o}{\partial h_4} \right] \cdot \frac{\partial L_0}{\partial o} = \frac{\partial L_0}{\partial o} \cdot \left[\frac{\partial o}{\partial h_4} \right] \cdot \frac{\partial h_4}{\partial w(h_3, h_4)}.
 \end{aligned}$$

In der Realität kann es jedoch exponentiell viele mögliche Pfade geben. Für den Fall, dass mehr als ein Pfad existiert, wird eine verallgemeinerte Variante der Kettenregel für die Berechnung der Ableitungen herangezogen, nämlich die mehrdimensionale Kettenregel. Dabei wird die Hintereinanderausführung entlang jedes Pfades von h_1 bis o hinzugefügt und obiger Ausdruck verallgemeinert sich auf den Fall, in dem eine Menge \mathcal{P} mit Pfaden von h_r bis o existiert:

$$\frac{\partial L}{\partial w(h_{r-1}, h_r)} = \frac{\partial L}{\partial o} \cdot \underbrace{\left[\sum_{[h_r, h_{r+1}, \dots, h_k, o] \in \mathcal{P}} \frac{\partial o}{\partial h_k} \prod_{i=r}^{k-1} \frac{\partial h_{i+1}}{\partial h_i} \right]}_{\text{Backpropagation berechnet } \Delta(h_r, o) = \frac{\partial L}{\partial h_r}} \cdot \frac{\partial h_r}{\partial w(h_{r-1}, h_r)}. \quad (4.2)$$

Der Term $\frac{\partial L}{\partial h_r}$ wird aufgrund der großen Anzahl an Pfaden rekursiv berechnet. Dies ist möglich, da ein Neuronales Netz ein azyklischer Graph ist. Zuerst wird $\frac{\partial L}{\partial h_r}$ für die Knoten

h_k berechnet, die am nächsten zu o liegen. Dann werden diese Werte für die anderen Knoten in den vorderen Layern rekursiv berechnet. Es gilt, dass $\Delta(o, o) = \frac{\partial L}{\partial o}$. Die Rekursion für $\frac{\partial L}{\partial h_r}$ erhält man mittels der mehrdimensionalen Kettenregel:

$$\Delta(h_r, o) = \frac{\partial L}{\partial h_r} = \sum_{h: h_r \Rightarrow h} \frac{\partial L}{\partial h} \frac{\partial h}{\partial h_r} = \sum_{h: h_r \Rightarrow h} \frac{\partial h}{\partial h_r} \Delta(h, o).$$

Da h in einem späteren Layer als h_r liegt, ist $\Delta(h, o)$ schon berechnet während $\Delta(h_r, o)$ evaluiert wird.

Die Berechnung von $\frac{\partial h}{\partial h_r}$ wird als nächstes behandelt. Hier wird die Situation betrachtet, dass h_r und h durch die Kante mit Gewicht $w(h_r, h)$ verbunden sind. Weiters sei a_h der Wert des Knoten h kurz bevor die Aktivierungsfunktion ϕ angewandt wird. Es gilt somit, dass $h = \phi(a_h)$, wobei a_h die Linearkombination der Inputs aus den vorigen Layern ist. Dann erhält man mittels der Kettenregel folgenden Ausdruck für $\frac{\partial h}{\partial h_r}$:

$$\frac{\partial h}{\partial h_r} = \frac{\partial h}{\partial a_h} \cdot \frac{\partial a_h}{\partial h_r} = \frac{\partial \phi(a_h)}{\partial a_h} \cdot w(h_r, h) = \phi'(a_h) \cdot w(h_r, h).$$

Einsetzen in die obige Formel liefert nun:

$$\Delta(h_r, o) = \sum_{h: h_r \Rightarrow h} \phi'(a_h) \cdot w(h_r, h) \cdot \Delta(h, o).$$

Mittels dieser Rekursion werden die Gradienten erfolgreich rückwärts berechnet und jeder Knoten wird genau einmal pro Durchlauf bearbeitet. Der übrige Term $\frac{\partial h_r}{\partial w(h_{r-1}, h_r)}$ auf der rechten Seite der Gleichung 4.2 lässt sich direkt berechnen:

$$\frac{\partial h_r}{\partial w(h_{r-1}, h_r)} = h_{r-1} \cdot \phi'(a_{h_r}).$$

4.5.2 (Stochastisches) Gradienten Abstiegsverfahren

Die Literatur für diesen Abschnitt stammt aus [GBC16, Chapter 4.3: Gradient-Based Optimization, Chapter 8: Optimization for Training Deep Models].

Backpropagation ist ein Algorithmus zur Berechnung der Gradienten. Das Gradienten Abstiegsverfahren ist ein Algorithmus, der folglich die Gewichte und Biases anpasst.

Mit dem Gradienten Abstiegsverfahren will man die Verlustfunktion L minimieren.

Angenommen es ist eine Funktion $y = f(x)$ gegeben, wobei sowohl x als auch y reellwertige Zahlen sind. Die Ableitung $f'(x)$ gibt die Steigung von $f(x)$ im Punkt x an. Anders ausgedrückt gibt sie an, wie man den Input anpassen muss, um die entsprechende Änderung im Output zu erhalten: $f(x + \epsilon) \approx f(x) + \epsilon f'(x)$.

Die *Richtungsableitung* in Richtung \mathbf{u} (Einheitsvektor) gibt die Steigung der Funktion f in Richtung u an. Für die Richtungsableitung gilt aufgrund der Kettenregel, dass $\frac{\partial}{\partial \alpha} f(\mathbf{x} + \alpha \mathbf{u}) = \mathbf{u}^\top \nabla_{\mathbf{x}} f(\mathbf{x})$, falls $\alpha = 0$. Um f zu minimieren, will man die Richtung finden, in der f am schnellsten fällt:

$$\min_{\mathbf{u}, \mathbf{u}^\top \mathbf{u}=1} \mathbf{u}^\top \nabla_{\mathbf{x}} f(\mathbf{x}).$$

Die Funktion f bzw. L wird verkleinert, indem man sich in Richtung des negativen Gradienten bewegt. Dies wird als *Gradienten Abstiegsverfahren* bezeichnet. Der steilste Abstieg gibt vor, wie die Gewichte im nächsten Schritt angepasst werden müssen:

$$\mathbf{x}_{neu} = \mathbf{x}_{alt} - \epsilon \nabla_{\mathbf{x}} f(\mathbf{x}_{alt}) \quad \text{bzw.} \quad \boldsymbol{\theta}_{neu} = \boldsymbol{\theta}_{alt} - \epsilon \nabla_{\boldsymbol{\theta}} L(f(\mathbf{x}; \boldsymbol{\theta}_{alt}), y),$$

wobei

- ϵ die Lernrate ist,
- $\boldsymbol{\theta}$ die Menge aller Gewichte und Biases ist,
- $f(\mathbf{x}; \boldsymbol{\theta})$ der vorhergesagte Output mit Input \mathbf{x} ist und
- y der tatsächliche Output ist.

Die Lernrate ist ein positiver Skalar, welcher die Schrittgröße bestimmt. Das Verfahren terminiert, wenn jeder Eintrag des Gradienten sehr nahe bei Null ist. Der Gradient von L wird zwar exakt bestimmt, dies benötigt allerdings zu viel Zeit, da alle Trainingsbeispiele in die Berechnung miteinfließen.

Daher verwenden die meisten Deep Learning Modelle das *Stochastische Gradienten Abstiegsverfahren*. Hierbei erhält man einen unverzerrten Schätzer des Gradienten, indem die Trainingsdaten zufällig gemischt und in sogenannte *Minibatches* $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ geteilt werden. Jeder Minibatch enthält nur eine geringe Anzahl an Trainingsbeispielen. Der Schätzer ergibt sich, indem der Gradient der Verlustfunktion pro Minibatch berechnet wird. Die resultierenden Gradienten werden aufsummiert und gemittelt. Das stochastische Verfahren ist somit um einiges schneller und die Gewichte werden folgendermaßen angepasst:

$$\boldsymbol{\theta}_{neu} = \boldsymbol{\theta}_{alt} - \epsilon \underbrace{\left(\frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L^{(i)}(f(\mathbf{x}^{(i)}; \boldsymbol{\theta}_{alt}), y^{(i)}) \right)}_{\text{Schätzer für Gradient}}.$$

Die Lernrate ist ein wesentlicher Parameter im stochastischen Gradienten Abstiegsverfahren. Bisher wurde es mit einer fixen Lernrate ϵ beschrieben. In der Praxis ist es notwendig diese schrittweise zu verkleinern.

4.5.3 Lernrate

Wie schon erwähnt wurde, will man die Lernrate nicht konstant wählen, da dies zu Problemen führen kann. Wählt man nämlich die Rate zu klein, dann benötigt der Algorithmus zu lange um überhaupt nahe an eine optimale Lösung zu gelangen. Wählt man auf der anderen Seite die Lernrate zu groß, dann kommt der Algorithmus zuerst relativ nahe an eine gute Lösung, danach würde er jedoch um den Punkt oszillieren oder sogar divergieren, falls

die hohe Rate beibehalten wird. Um diese Probleme zu umgehen, will man eine Lernrate wählen, die über die Zeit abnimmt. Man will zu Beginn eine höhere Lernrate festsetzen und je näher man an die optimale Lösung kommt, desto kleiner soll die Lernrate werden. [A⁺18, Chapter 3.5.1: Learning Rate Decay]

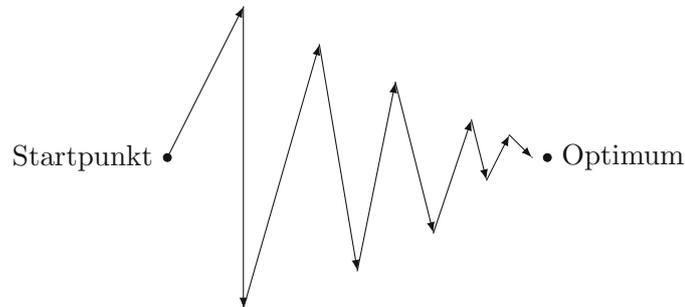


Abbildung 4.4: Abfallende Lernrate ohne Momentum

4.5.4 Momentum

Das Momentum-basierte Lernen wird aus [GBC16, Chapter 8.3.2: Momentum] und [A⁺18, Chapter 3.5.2 Momentum-based Learning] entnommen.

Das Stochastische Gradienten Abstiegsverfahren ist eine beliebte Optimierungsstrategie, allerdings verlangsamt es sich in flachen Regionen und bleibt bei lokalen Optima hängen. Die Momentum Methode beschleunigt den Lernprozess indem sie diese Verlangsamungen und lokalen Fallen vermeidet.

Momentum-basierte Techniken erkennen, dass dieser Zickzackverlauf die effektive Schrittgröße in die richtige Richtung reduziert. Es macht somit mehr Sinn in eine „gemittelte“ Richtung der letzten paar Schritte zu gehen, sodass der Zickzackverlauf geglättet wird.

Für den Momentum Algorithmus wird ein Vektor $\boldsymbol{\nu}$ definiert, die Velozität (engl: *velocity*), welche die Geschwindigkeit und Richtung vorgibt, mit welcher sich die Parameter durch den Parameterraum bewegen. Der Hyperparameter $\alpha \in [0, 1)$ bestimmt, wie schnell die vorherigen Gradienten exponentiell abfallen. Das Update ist dann gegeben durch:

$$\boldsymbol{\nu}_{t+1} = \alpha \boldsymbol{\nu}_t - \epsilon \nabla_{\boldsymbol{\theta}} \left(\frac{1}{m} \sum_{i=1}^m L^{(i)}(f(\mathbf{x}^{(i)}; \boldsymbol{\theta}_t), \mathbf{y}^{(i)}) \right)$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \boldsymbol{\nu}_{t+1}.$$

Je größer α im Verhältnis zu ϵ ist, desto mehr beeinflussen die vorherigen Gradienten die momentane Richtung.

Wenn man Abbildung 4.5 mit Abbildung 4.4 vergleicht, sieht man den glättenden Effekt der Momentum Methode auf den Zickzackverlauf. Der Lernprozess wird beschleunigt, da

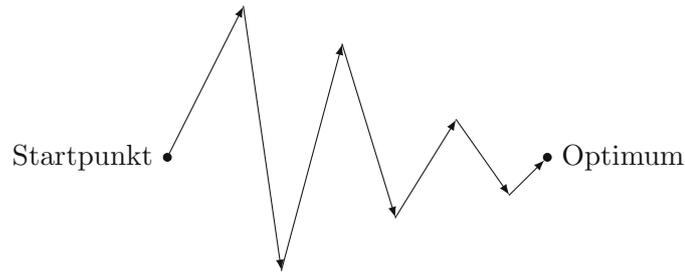


Abbildung 4.5: Abfallende Lernrate mit Momentum

man sich in eine Richtung bewegt, die näher zu der optimalen Lösung zeigt und Umwege reduziert werden.

4.5.5 Algorithmen mit adaptiver Lernrate

Die verschiedenen Algorithmen mit adaptiver Lernrate stammen aus [GBC16, Chapter 8.5: Algorithms with Adaptive Learning Rates], [A⁺18, Chapter 3.5.3: Parameter-Specific Learning Rates] und [Rud16].

Die Idee ist, dass Parameter mit großen partiellen Ableitungen oft oszillieren und einen Zickzackverlauf haben und Parameter mit kleineren partiellen Ableitungen konsistenter sind, sich allerdings in dieselbe Richtung bewegen. Man will nun die Lernrate in Abhängigkeit der partiellen Ableitungen wählen und außerdem für verschiedene Parameter verschiedene Lernraten definieren.

AdaGrad

Der AdaGrad Algorithmus adaptiert die Lernrate eines jeden Modellparameters, indem die globale Lernrate invers proportional zur Quadratwurzel der Summe aller vergangenen Gradienten zum Quadrat skaliert wird. Parameter mit großen partiellen Ableitungen der Verlustfunktion weisen einen schnellen Abfall der Lernrate und Parameter mit kleinen partiellen Ableitungen nur eine kleine Abnahme in der jeweiligen Lernrate auf.

Für jeden Updateschritt benötigt man:

1. den Gradienten $\mathbf{g}_t = \nabla_{\boldsymbol{\theta}} \left(\frac{1}{m} \sum_{i=1}^m L^{(i)}(f(\mathbf{x}^{(i)}; \boldsymbol{\theta}_t), \mathbf{y}^{(i)}) \right)$,
2. $\mathbf{r}_t = \mathbf{r}_{t-1} + \mathbf{g}_t \odot \mathbf{g}_t$, wobei \mathbf{r} zu Beginn mit $\mathbf{r} = \mathbf{0}$ initialisiert wird und \odot die komponentenweise Multiplikation darstellt,
3. die globale Lernrate ϵ und
4. eine kleine Konstante δ , beispielsweise 10^{-8} , welche eine Division durch Null verhindert.

Die Updaterregel des AdaGrad Algorithmus ist dann gegeben durch:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \underbrace{\frac{\epsilon}{\delta + \sqrt{\mathbf{r}_t}}}_{\substack{\text{adaptive} \\ \text{Lernrate}}} \odot \mathbf{g}_t.$$

Division, Multiplikation und Wurzel werden elementweise angewandt. Die Kumulierung der quadratischen Gradienten von Beginn an kann zu einem verfrühten und starkem Abfall der effektiven Lernrate führen. AdaGrad konvergiert schnell für konvexe Funktionen und performt somit gut für einige, aber nicht alle Deep Learning Modelle.

RMSProp

Der RMSProp Algorithmus ist eine modifizierte Version des AdaGrad Algorithmus. Er weist in einem nicht-konvexen Setting eine bessere Performance auf, da die Kumulierung der quadratischen Gradienten etwas abgeändert wird. Die Idee ist es einen Abklingfaktor (engl.: *decay factor*) $\rho \in (0, 1)$ zu verwenden und die partiellen Ableitungen zum Quadrat, welche vor t Updates vorkommen, mit ρ^t zu gewichten.

Für jeden Updateschritt benötigt man wiederum:

1. den Gradienten $\mathbf{g}_t = \nabla_{\boldsymbol{\theta}} \left(\frac{1}{m} \sum_{i=1}^m L^{(i)}(f(\mathbf{x}^{(i)}; \boldsymbol{\theta}_t), \mathbf{y}^{(i)}) \right)$,
2. $\mathbf{r}_t = \rho \cdot \mathbf{r}_{t-1} + (1 - \rho) \mathbf{g}_t \odot \mathbf{g}_t$, wobei \mathbf{r} zu Beginn mit $\mathbf{r} = \mathbf{0}$ initialisiert wird,
3. die globale Lernrate ϵ ,
4. eine kleine Konstante δ , meistens 10^{-6} , welche eine Division durch Null verhindert und
5. eine Abklingrate $\rho \in (0, 1)$, welche oft als $\rho = 0.9$ gewählt wird

Die Updaterregel ist gegeben durch:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \underbrace{\frac{\epsilon}{\sqrt{\delta + \mathbf{r}_t}}}_{\substack{\text{adaptive} \\ \text{Lernrate}}} \odot \mathbf{g}_t.$$

Alle Operationen werden wieder elementweise angewandt. Der Vorteil des RMSProp gegenüber AdaGrad ist, dass die Bedeutung alter Gradienten über Zeit exponentiell abnimmt, da beispielsweise $\mathbf{g}_1 \odot \mathbf{g}_1$ nach t Schritten mit ρ^t gewichtet ist und $\rho \in (0, 1)$. Damit gilt, je größer t , desto geringer ist der Einfluss der früheren Gradienten zum Quadrat.

RMSProp ist somit ein effektiver und häufig verwendeter Algorithmus für die Optimierung Neuronaler Netze.

Adam

Der Name Adam leitet sich aus dem Ausdruck „adaptive Momente“ ab und ist eine Kombination des RMSProp Algorithmus und dem Momentum.

Zusätzlich zu der Eigenschaft des RMSProp Algorithmus, dass die Bedeutung der vergangenen Gradienten zum Quadrat \mathbf{r}_t mit der Zeit abnimmt, berücksichtigt Adam auch die exponentielle Abnahme der vergangenen Gradienten \mathbf{s}_t , ähnlich wie der Momentum Algorithmus.

Für jeden Updateschritt benötigt man:

1. die Schrittgröße bzw. die globale Lernrate ϵ (oft gewählt als 0.001),
2. die exponentiellen Abklingraten ρ_1 und $\rho_2 \in [0, 1)$ für die Schätzer der Momente (oft gewählt als 0.9 bzw. 0.999),
3. eine kleine Konstante δ (z.B. 10^{-8}), um eine Division durch Null zu vermeiden,
4. den Gradienten $\mathbf{g}_t = \nabla_{\boldsymbol{\theta}} \left(\frac{1}{m} \sum_{i=1}^m L(f(\mathbf{x}^{(i)}; \boldsymbol{\theta}_t), \mathbf{y}^{(i)}) \right)$,
5. den verzerrten Schätzer für das erste Moment ($\mathbb{E}[\mathbf{g}_t]$): $\mathbf{s}_t = \rho_1 \mathbf{s}_{t-1} + (1 - \rho_1) \mathbf{g}_t$ und den verzerrten Schätzer für das zweite Moment ($\mathbb{V}[\mathbf{g}_t]$): $\mathbf{r}_t = \rho_2 \mathbf{r}_{t-1} + (1 - \rho_2) \mathbf{g}_t \odot \mathbf{g}_t$, wobei $\mathbf{s} = \mathbf{0}$, $\mathbf{r} = \mathbf{0}$ und $t = 0$ zu Beginn gesetzt werden,
6. den Schätzer für das erste Moment mit korrigiertem Bias $\hat{\mathbf{s}}_t = \frac{\mathbf{s}_t}{1 - \rho_1^t}$ und den Schätzer für das zweite Moment mit korrigiertem Bias $\hat{\mathbf{r}}_t = \frac{\mathbf{r}_t}{1 - \rho_2^t}$.

Die Updateregeln lautet:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \epsilon \frac{\hat{\mathbf{s}}_t}{\sqrt{\hat{\mathbf{r}}_t + \delta}}.$$

Auch hier werden alle Operationen elementweise angewandt. Der Adam Algorithmus zählt zu den am meisten verwendeten Algorithmen für die Optimierung Neuronaler Netzwerke.

Für alle vorgestellten Algorithmen gilt, dass die Updateschritte auf Minibatches erfolgen und so lange durchgeführt werden, so lange das Abbruchkriterium nicht erfüllt ist.

5 Beispiel

In diesem Kapitel wird ein Datensatz generiert, der zwei stochastische Komponenten, welche das Langlebigkeitsrisiko darstellen, enthält. Danach wird ein Deep Learning Modell mithilfe dieser Daten trainiert, welches schlussendlich den Betrag der Rückversicherung vorhersagen soll. Anschließend werden die Daten der „AVÖ 2018-P - Rechnungsgrundlagen für Pensionsversicherung“ [KHS18] herangezogen und Vorhersagen mittels des trainierten Modells gemacht.

5.1 Datensatzerstellung

Für die Generierung der Daten wird das stochastische Gompertz-Makeham Modell mit Ornstein-Uhlenbeck Prozessen mit $k = 1$ aus [Rhe19] herangezogen, da in diesem Modell die Parameter zufällig beschrieben werden. Für die Sterbeintensität gilt:

$$\mu_x(t) = Y_1(t) + Y_2(t) \cdot \exp(\gamma x).$$

Hier sind Y_1 und Y_2 zwei Ornstein-Uhlenbeck Prozesse mit $k = 1$, welche die stochastischen Differentialgleichungen lösen:

$$\begin{aligned} dY_1(t) &= (\mu - Y_1(t)) dt + \sigma dW_t^1 \\ dY_2(t) &= (\mu - Y_2(t)) dt + \sigma dW_t^2. \end{aligned}$$

Es gilt, dass $\mu, \sigma > 0$ und W^1, W^2 Brownsche Bewegungen sind.

Im ersten Schritt sind die Lösungen der beiden stochastischen Differentialgleichungen gesucht. Diese erhält man durch Anwendung des Euler-Maruyama Algorithmus, welcher in [Hig01] beschrieben wird.

Gegeben ist eine Differentialgleichung der Form

$$dX(t) = f(X(t))dt + g(X(t))dW_t, \quad X(0) = X_0, \quad 0 \leq t \leq T.$$

Um eine numerische Methode auf obige Gleichung über $[0, T]$ anzuwenden, muss zuerst das Intervall diskretisiert werden. Dafür sei $\Delta t = T/L$ für eine positive ganze Zahl L und

$\tau_j = j\Delta t$. Die numerische Approximation für $X(\tau_j)$ wird mit X_j bezeichnet. Die Lösung mittels Euler Methode ist dann folgende:

$$X_j = X_{j-1} + f(X_{j-1})\Delta t + g(X_{j-1})(W(\tau_j) - W(\tau_{j-1})), \quad j = 1, 2, \dots, L.$$

Die Lösung der obigen beiden SDEs sieht dann folgendermaßen aus, wobei dt die Schrittweite ist und die Inkremente der Brownschen Bewegung normalverteilt sind:

$$\begin{aligned} \hat{Y}_i(0) &= Y_i(0) \\ \hat{Y}_i(t+1) &= \hat{Y}_i(t) + (\mu - \hat{Y}_i(t))dt + \sigma\mathcal{N}(0, \sqrt{dt}), \quad \mu > 0, \sigma > 0, i = 1, 2. \end{aligned}$$

Damit erhält man μ_x und die zugehörige Überlebenswahrscheinlichkeit hat folgende Form:

$${}_{T-t}p_{x+t}(t) = E[\exp(-\int_t^T \mu_{x+s}(s)ds) | \mathcal{F}_t].$$

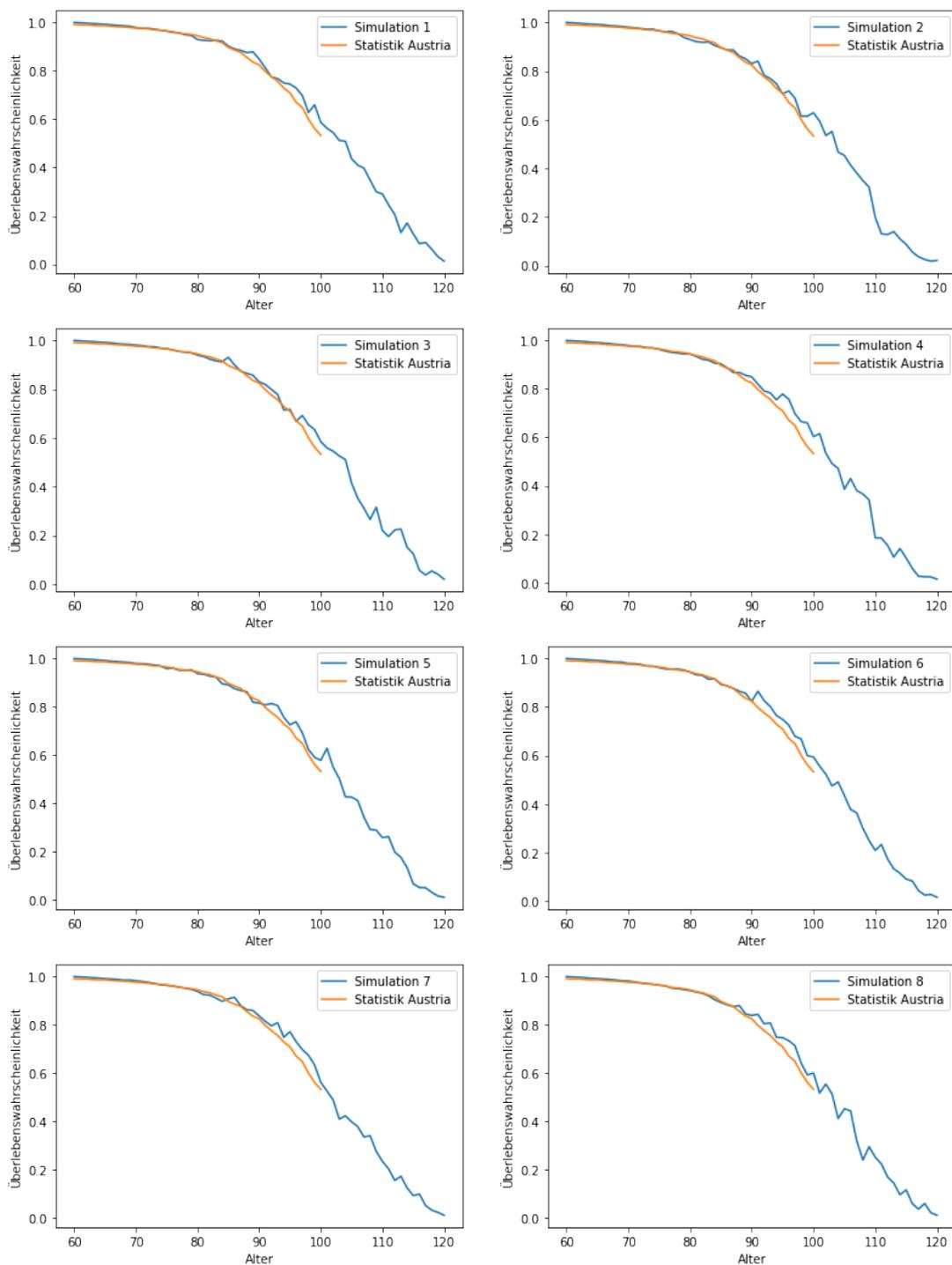
Für die explizite Berechnung der Überlebenswahrscheinlichkeiten p_x wird die Survivalfunktion aus [Nor97] herangezogen und die Parameter stochastisch als Ornstein-Uhlenbeck Prozesse gewählt. Die Gompertz-Makeham Verteilungsfunktion liefert sehr gute Approximationen der Sterbetafeln. Sie verwendet die einfache Gompertzfunktion und enthält mehr Parameter als diese. Die Gompertz-Makeham Verteilung hat die Survivalfunktion:

$$S(t) = \exp\left(-\alpha t - \beta \frac{e^{\gamma t} - 1}{\gamma}\right),$$

wobei hier $\alpha = Y_1(t)$ und $\beta = Y_2(t)$ gewählt werden.

Die Parameter μ , σ und γ werden so gewählt, dass die Survivalfunktion die Überlebenswahrscheinlichkeiten der Statistik Austria ¹ approximiert. Hierfür wurden die Daten der „Sterbetafel 2021 für Österreich“ verwendet. Es wird zwischen Männern und Frauen unterschieden und jeweils acht Simulationen der Überlebenswahrscheinlichkeiten durchgeführt. Die simulierten Wahrscheinlichkeiten bilden die Grundlage des Datensatzes, da mittels diesen die Rentenbarwerte berechnet werden und folglich die Anteile des Erstversicherers sowie die des Rückversicherers festgelegt werden. Da hier Rentenbarwerte ab einem Alter von 60 Jahren für Männer und Frauen berechnet werden, betrachtet man nur die Überlebenswahrscheinlichkeiten für das Alter 60 bis 120.

¹<https://www.statistik.at/statistiken/bevoelkerung-und-soziales/bevoelkerung/demographische-indikatoren-und-tafeln/sterbetafeln>

Abbildung 5.1: Simulationen Männer mit $\mu = 0.00069$, $\sigma = 0.0001$ und $\gamma = 0.108$

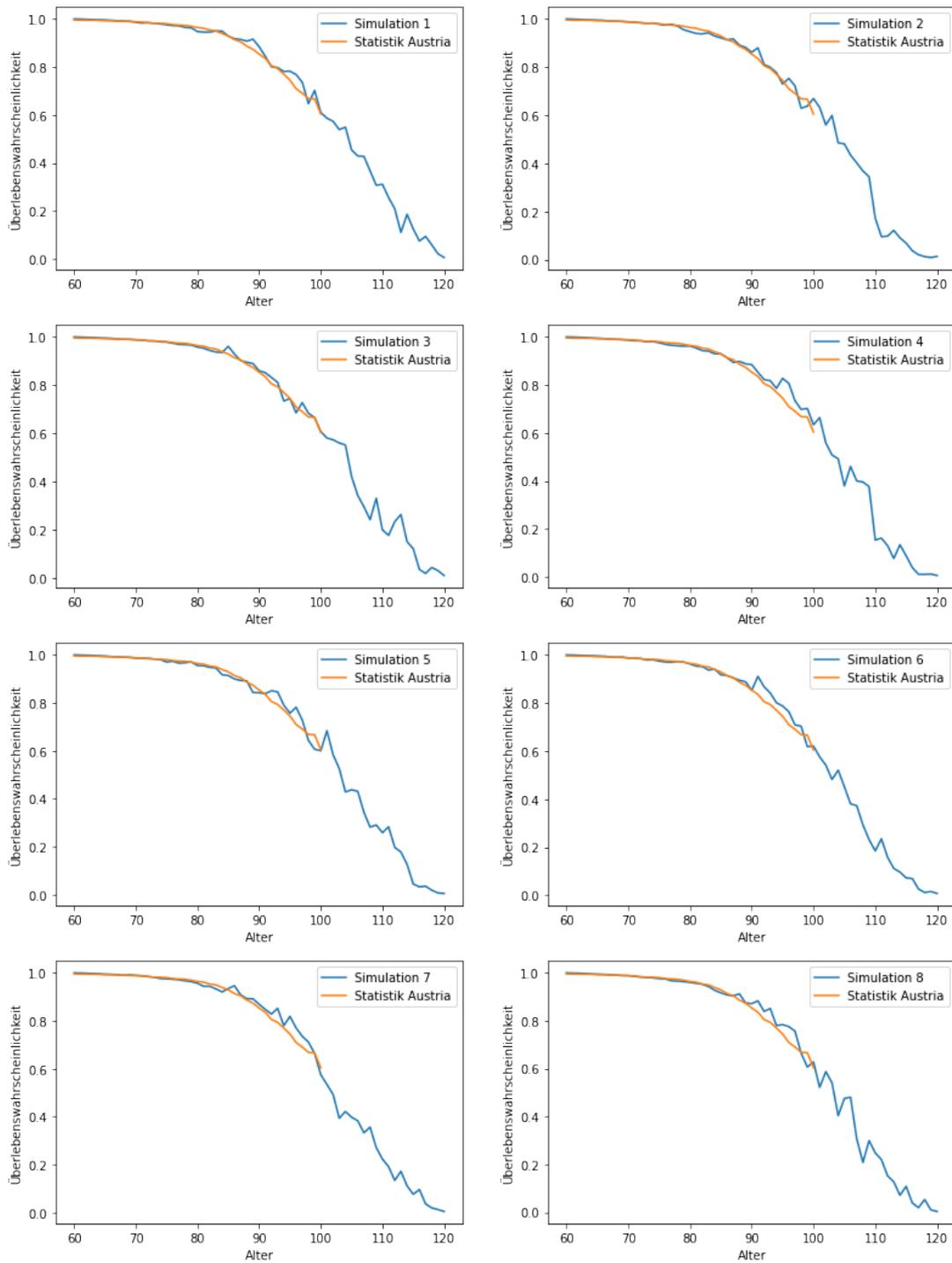


Abbildung 5.2: Simulationen Frauen mit $\mu = 0.000418$, $\sigma = 0.00009$ und $\gamma = 0.1211$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Mehrmalige Auswertungen des Modells haben ergeben, dass mit einer leichten Überschätzung der simulierten Überlebenswahrscheinlichkeiten bessere Ergebnisse erzielt werden konnten. Vermutlich sorgten negative Ausreißer der Überlebenswahrscheinlichkeiten dafür, dass die Berechnungen verzerrt wurden.

Es werden lebenslange Leibrenten wie in [SSK09] betrachtet. Dabei handelt es sich um Leibrenten, die einmal jährlich Zahlungen der Höhe 1 leisten. Durch Multiplikation mit der tatsächlichen Rente können die erhaltenen Formeln auf den allgemeineren Fall übertragen werden. Prinzipiell ist zwischen vor- und nachschüssigen Leibrenten zu unterscheiden, wobei in dem Beispiel erstere verwendet wird. Eine *vorschüssige lebenslängliche Leibrente* leistet Zahlungen der Höhe 1 zu Beginn eines jeden Jahres solange die versicherte Person am Leben ist. Somit gilt für den Barwert einer solchen Leibrente \ddot{a}_x :

$$\ddot{a}_x = \sum_{k=0}^{\infty} v^k {}_k p_x = \sum_{k=0}^{\infty} v^k \prod_{i=0}^{k-1} p_{x+i},$$

wobei

- v der Diskontierungsfaktor ist und
- ${}_k p_x$ die Wahrscheinlichkeit, dass eine x -jährige Person die nächsten k Jahre überlebt ist.

Letztendlich werden folgende Annahmen an den Datensatz getroffen:

- Der Datensatz besteht aus Männern (gekennzeichnet mit 0) und Frauen (gekennzeichnet mit 1) im Alter von 60 bis 120 Jahren.
- Die Überlebenswahrscheinlichkeiten, mittels welchen man die Rentenbarwerte berechnet, enthalten zwei stochastische Komponenten, die das Langlebighkeitsrisiko darstellen.
- In der Berechnung der Rentenbarwerte \ddot{a}_x wird ein Zinssatz von 2% gewählt.
- Als Rückversicherungsart wird die XL- Rückversicherung gewählt. Der Selbstbehalt des Erstversicherers wird mit 15 EUR festgelegt.
- Die Spalten Alter, Geschlecht und p_x stellen den Input für das Modell dar. Mittels diesen Informationen soll der Anteil der Rückversicherung vorhergesagt werden.

Der generierte Datensatz wird in den folgenden beiden Tabellen dargestellt. Es handelt sich um eine Simulation von acht, wobei der erste Datensatz die Daten der Männer enthält und der zweite die Daten der Frauen. Der gesamte Datensatz besteht letztendlich aus den Daten aller acht Simulationen beider Geschlechter.

Tabelle 5.1: Datensatz Männer mit Wahrscheinlichkeiten aus Simulation 1

Alter	G	p_x	\ddot{a}_x	EV	RV
60	0	1.0	18.62211404877403	15	3.62211404877403
61	0	0.9986649298535801	17.97455632974951	15	2.974556329749511
62	0	0.9970178403447228	17.337193826244622	15	2.337193826244622
63	0	0.9954383980499201	16.713780865754515	15	1.713780865754515
64	0	0.9941635677101239	16.10150514031689	15	1.1015051403168883
65	0	0.9927033480390711	15.493964719108037	15	0.4939647191080372
66	0	0.9903115732557314	14.89250947193172	15	0.0
67	0	0.98791077766961	14.308991275123773	15	0.0
68	0	0.9862839770888027	13.741292642488244	15	0.0
69	0	0.9844068331153495	13.176852506210656	15	0.0
70	0	0.9791655833791164	12.617130579059578	15	0.0
⋮	⋮	⋮	⋮	⋮	⋮
80	0	0.9297762771863056	7.871103814462713	15	0.0
81	0	0.9260975203487889	7.537862669459807	15	0.0
82	0	0.924277762158258	7.20077505481005	15	0.0
83	0	0.926222511946284	6.842954374599897	15	0.0
84	0	0.922747692287097	6.434537473688106	15	0.0
85	0	0.9022925495035111	6.007306514549575	15	0.0
86	0	0.8904168620799856	5.660528447952895	15	0.0
87	0	0.8845658947049216	5.338779193609799	15	0.0
88	0	0.8753134291391582	5.003080950750759	15	0.0
89	0	0.8786703381660216	4.664777705720121	15	0.0
90	0	0.8487532058608209	4.254238589226427	15	0.0
⋮	⋮	⋮	⋮	⋮	⋮
100	0	0.5871472952123904	2.2203055840745014	15	0.0
101	0	0.5626588273287193	2.1199309030381195	15	0.0
102	0	0.5449877750759369	2.0302347810345514	15	0.0
103	0	0.5124173011021012	1.9281890800372963	15	0.0
104	0	0.5087612672658358	1.8476207957884663	15	0.0
105	0	0.43652818623808	1.6993691684718646	15	0.0
106	0	0.4102200969864374	1.6341592005521526	15	0.0
107	0	0.3976196448970188	1.576817882193084	15	0.0
108	0	0.3487652710984528	1.479691075095964	15	0.0
109	0	0.2999410473652426	1.4029060148588106	15	0.0
110	0	0.2910539042892577	1.3701496969687852	15	0.0
⋮	⋮	⋮	⋮	⋮	⋮
120	0	0.013600666430498218	1.013333986696567	15	0.0

Tabelle 5.2: Datensatz Frauen mit Wahrscheinlichkeiten aus Simulation 1

Alter	G	p_x	\ddot{a}_x	EV	RV
60	1	1.0	20.2805548671933	15	5.280554867193299
61	1	0.9992132737434494	19.666165964537182	15	4.666165964537182
62	1	0.9981588380268603	19.05447994350439	15	4.05447994350439
63	1	0.9971850564597802	18.449538130402168	15	3.4495381304021677
64	1	0.996513575243589	17.84877217895621	15	2.84877217895621
65	1	0.9956818378842703	17.24587406482087	15	2.245874064820871
66	1	0.9940311234043232	16.642657238107976	15	1.6426572381079758
67	1	0.9923599497873169	16.05131872352876	15	1.0513187235287589
68	1	0.9915183150087409	15.470540806578953	15	0.4705408065789527
69	1	0.9903727021032654	14.886211781756563	15	0.0
70	1	0.9861072876675367	14.301621992722128	15	0.0
:	:	:	:	:	:
80	1	0.9473961555098918	9.096958349758353	15	0.0
81	1	0.9449734360739377	8.71746995036997	15	0.0
82	1	0.9452158894696692	8.330201727238245	15	0.0
83	1	0.9499508882518166	7.910156658473031	15	0.0
84	1	0.949000021724404	7.4197096700583165	15	0.0
85	1	0.9280580928817237	6.900003913130673	15	0.0
86	1	0.9175194357693838	6.484512163141335	15	0.0
87	1	0.9147034256993841	6.097094174046741	15	0.0
88	1	0.9075503704119592	5.683848897310601	15	0.0
89	1	0.9155367540509768	5.264199135402455	15	0.0
90	1	0.8843207573339436	4.750746596316687	15	0.0
:	:	:	:	:	:
100	1	0.6100623916052893	2.333496026367775	15	0.0
101	1	0.5863287408456007	2.2295521992694116	15	0.0
102	1	0.573286049541157	2.1389762361744036	15	0.0
103	1	0.5389746936703345	2.026485315363473	15	0.0
104	1	0.5490614242217149	1.9426051611824877	15	0.0
105	1	0.45412494606777687	1.751092358690081	15	0.0
106	1	0.4291291111033998	1.6870119391096872	15	0.0
107	1	0.42711954286005516	1.632963506227926	15	0.0
108	1	0.3671558732870068	1.5115739542829143	15	0.0
109	1	0.3070570506484362	1.4212095497670985	15	0.0
110	1	0.31109742885962005	1.3991984221015272	15	0.0
:	:	:	:	:	:
120	1	0.005875749664438189	1.0057605388867041	15	0.0

Es ergeben sich insgesamt 976 Datenpunkte, die anschließend in 80% Trainingsdaten und 20% Validierungsdaten aufgeteilt werden. Das Modell wird dann mittels Trainingsdatensatz trainiert und mittels Validierungsdatensatz evaluiert.

5.2 Deep Learning Modelle

Für die Erstellung der Modelle wird Python mit der Deep-Learning Bibliothek Keras verwendet. Aus den Arbeiten [Ger19] und [Bau19] wird entnommen, dass ein Modell mit drei hidden Layern, jeweils mit 64 Knoten und mit Tanh als Aktivierungsfunktion, und dem Adam Optimizer eine gute Wahl für die jeweiligen Problemstellungen war. Deshalb wird auch für den erzeugten Datensatz dieses Modell verwendet und analysiert. Anschließend werden Verbesserungen an dem Modell vorgenommen und weitere Modelle vorgestellt, welche eine bessere Performance vorweisen.

5.2.1 Tanh Modelle

Modell 1

Zuerst wird ein Modell mit 3 hidden Layern mit jeweils 64 Knoten und Tangens hyperbolicus als Aktivierungsfunktion betrachtet. Die Anzahl der Epochen beträgt 800. In allen Modellen wird der Adam Optimizer verwendet, da dieser die besten Ergebnisse liefert. Der Defaultwert für die Lernrate des Adam Optimizers ist 0.001 und wird in diesem Modell verwendet.

Das Modell schneidet schon relativ gut ab, denn bei der Evaluierung der Validierungsdaten erhält man einen MAE von 0.0325 und einen MSE von 0.0027.

Im linken Plot in Abbildung 5.3 wird der MSE des Trainingsdatensatzes (blau) und des Validierungsdatensatzes (orange) für alle Epochen abgebildet. Man sieht, dass sich Trainings- und Validierungsfehler etwas unterscheiden, allerdings sind beide auf niedrigem Niveau. Der rechte Plot zeigt die Differenz zwischen den tatsächlichen und vorhergesagten Werten für den Rückversicherungsbetrag. Hier lässt sich ablesen, dass die Unterschiede für den Großteil der Beobachtungen relativ klein sind. Es gibt vereinzelt Ausreißer und die größte Differenz beträgt ungefähr 0.30 EUR.

Es ist jedoch möglich das Modell noch weiter zu verbessern, indem man Änderungen an einigen Hyperparametern vornimmt.

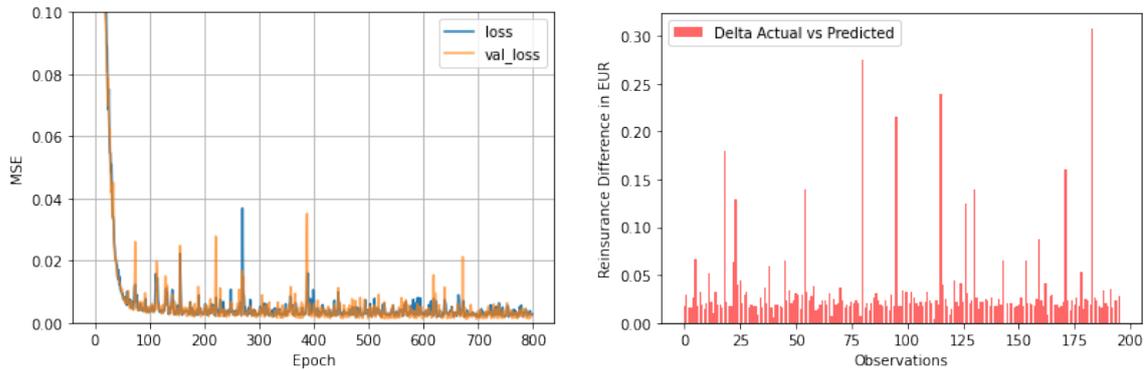


Abbildung 5.3: Tanh Modell mit 3 hidden Layern mit jeweils 64 Knoten und 800 Epochen

Modell 2

In diesem Modell wird die Anzahl der hidden Layer auf 4 und die Anzahl der Knoten auf 512 erhöht. Dabei enthält der erste dieser Layer 256, der zweite Layer 128 und die letzten beiden Layer jeweils 64 Knoten. Weiters wird für die Lernrate der Wert 0.0001 gewählt und ist somit kleiner als in Modell 1. Damit will man die Ausreißer im linken Plot in Abbildung 5.3 glätten. Dieses Modell liefert für die Validierungsdaten einen MAE von 0.0173 und einen MSE von 0.0017. Dies ist eine Verbesserung zu Modell 1, da sich aus den Plots in Abbildung 5.4 erkennen lässt, dass sowohl Trainings- als auch Validierungsfehler für die einzelnen Epochen kleiner geworden sind und die größte Differenz 0.20 EUR beträgt.

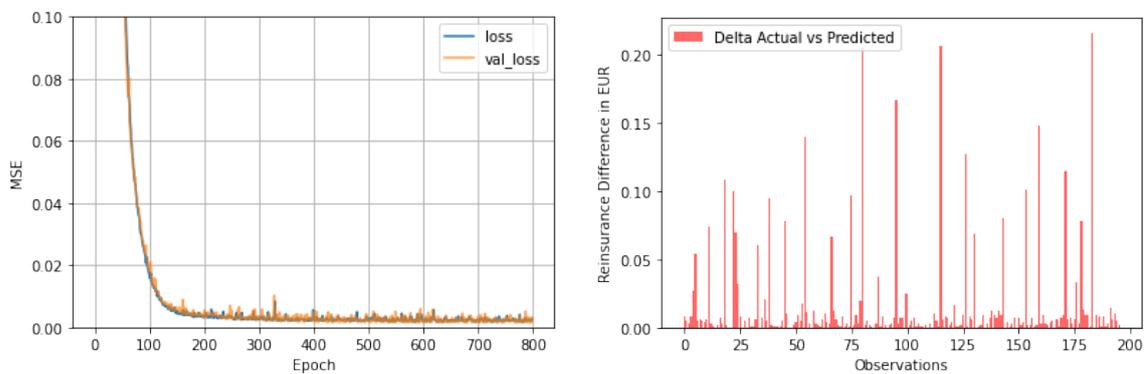


Abbildung 5.4: Tanh Modell mit 4 hidden Layern, 512 ($1 \times 256, 1 \times 128, 2 \times 64$) Knoten und 800 Epochen

Ein letzter Versuch das Modell zu verbessern besteht darin die Trainingsepochen zu erhöhen.

Modell 3

Dieses Modell ist ident zu Modell 2, es wird nur die Anzahl der Epochen auf 1200 erhöht. Die Evaluierung der Validierungsdaten liefert hier einen MAE von 0.0137 und einen MSE

von 0.0012. Diese Änderung liefert nochmals eine deutliche Verbesserung für MAE und MSE. Trainings- und Validierungsfehler im linken Plot in Abbildung 5.5 verhalten sich gleich wie in Modell 2, allerdings beträgt der größte Unterschied zwischen tatsächlichen und vorhergesagten Rückversicherungsbeträgen nur noch 0.175 EUR.

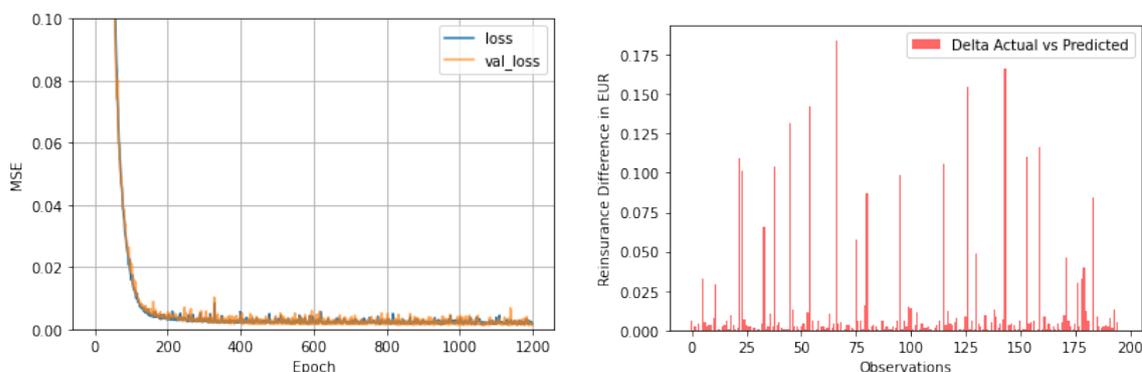


Abbildung 5.5: Tanh Modell 2 mit 1200 Epochen

Modell 3 liefert soweit die besten Ergebnisse. Man kann jedoch diese Modelle ebenso verbessern, indem man eine andere Aktivierungsfunktion wählt.

5.2.2 ReLU Modelle

Rectified Linear Unit ist eine weitere sehr beliebte Wahl für die Aktivierungsfunktion. In vielen Fällen können mit dieser Wahl bessere Ergebnisse erzielt werden. In diesem Abschnitt wird versucht die obigen Tanh Modelle zu verbessern, indem ReLU als Aktivierungsfunktion gewählt wird.

Modell 1

Zu Beginn wird ein Modell mit wiederum 3 hidden Layern mit jeweils 64 Knoten und ReLU als Aktivierungsfunktion betrachtet. Die Anzahl der Epochen wird auf 800 gesetzt. Auch bei den ReLU-Modellen wird Adam als Optimizer verwendet und in Modell 1 die Default Lernrate gewählt.

Dieses Modell liefert bei der Evaluierung der Validierungsdaten einen MAE von 0.0131 und einen MSE von 0.0018. In Abbildung 5.6 erkennt man im linken Plot, dass sich Trainings- und Validierungsfehler wie auch im Tanh Modell 1 leicht unterscheiden, allerdings relativ klein sind. Im rechten Plot sieht man, dass die Differenz zwischen tatsächlichen und vorhergesagten Rückversicherungsbeträgen bei den meisten Beobachtungen auf sehr kleinem Niveau ist. Es gibt lediglich ein paar Ausreißer und die größte Differenz beträgt 0.25 EUR.

In diesem Abschnitt wird ebenso versucht die einzelnen ReLU Modelle schrittweise zu verbessern, indem man einige Hyperparameter abändert.

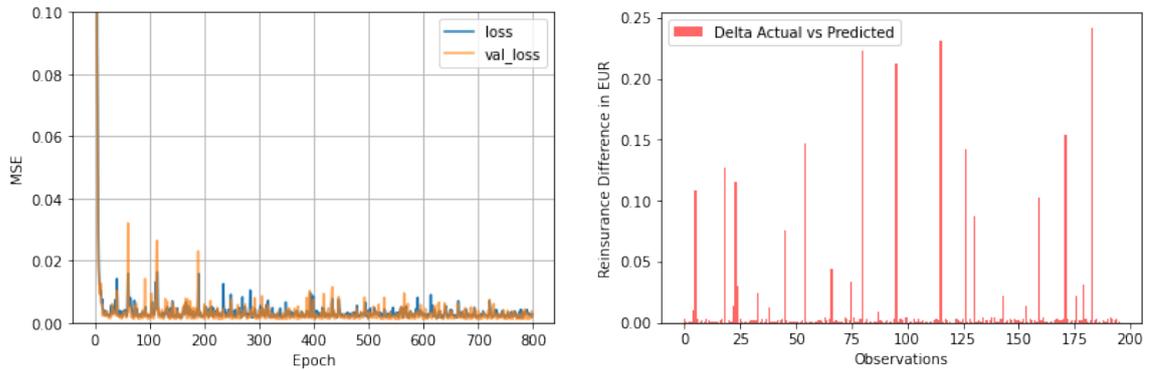


Abbildung 5.6: ReLU Modell mit 3 hidden Layern mit jeweils 64 Knoten und 800 Epochen

Modell 2

In diesem Modell wird die Anzahl der hidden Layer und Knoten erhöht und die Lernrate mit 0.0001 kleiner gewählt also zuvor. ReLU Modell 2 besitzt 4 hidden Layer, wobei der erste dieser Layer 256 Knoten, der zweite und dritte Layer jeweils 128, der letzte Layer 64 Knoten enthält. Dies liefert einen MAE von 0.0120 und einen MSE von 0.0012.

In Abbildung 5.7 im linken Plot lässt sich die glättende Wirkung der kleineren Lernrate gut erkennen. Der Unterschied zwischen Trainings- und Validierungsfehler ist ebenfalls geringer geworden. Der rechte Plot zeigt, dass die größte Differenz in den Beobachtungen nur noch 0.175 EUR beträgt. Dies stellt eine deutliche Verbesserung zu ReLU Modell 1 dar.

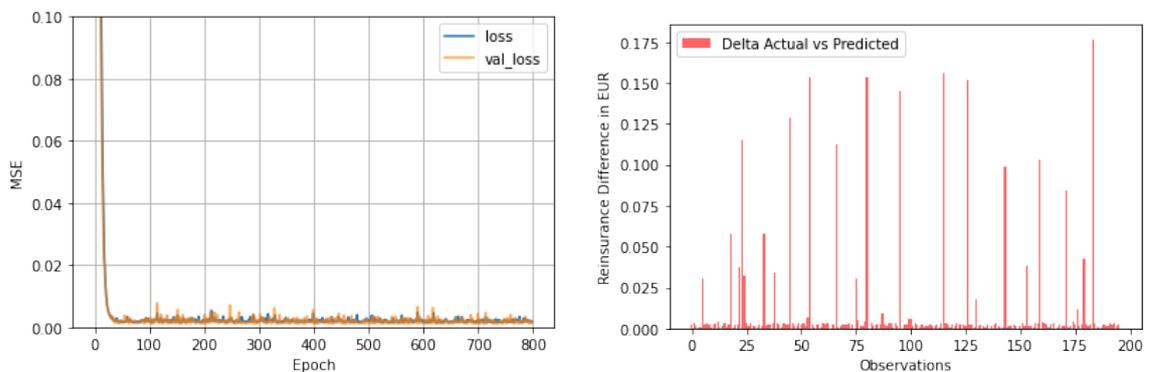


Abbildung 5.7: ReLU Modell mit 4 hidden Layern, 576 ($1 \times 256, 2 \times 128, 1 \times 64$) Knoten und 800 Epochen

Im letzten Versuch wird wiederum die Anzahl der Epochen erhöht. Dieses Modell liefert die bisher besten Ergebnisse mit den geringsten Fehlern für die Validierungsdaten.

Modell 3

Das letzte Modell ist ident zu ReLU Modell 2, die Epochen werden jedoch auf 1200 erhöht. Die Evaluierung der Validierungsdaten liefert damit einen MAE von 0.0110 und einen MSE von 0.0011. Diese Erhöhung der Epochen führt somit zu einer Verbesserung der Ergebnisse. In Abbildung 5.8 im rechten Plot erkennt man zwar, dass die größte Differenz 0.20 EUR beträgt, allerdings sind die restlichen Differenzen in Modell 3 niedriger verglichen mit denen in Modell 2.

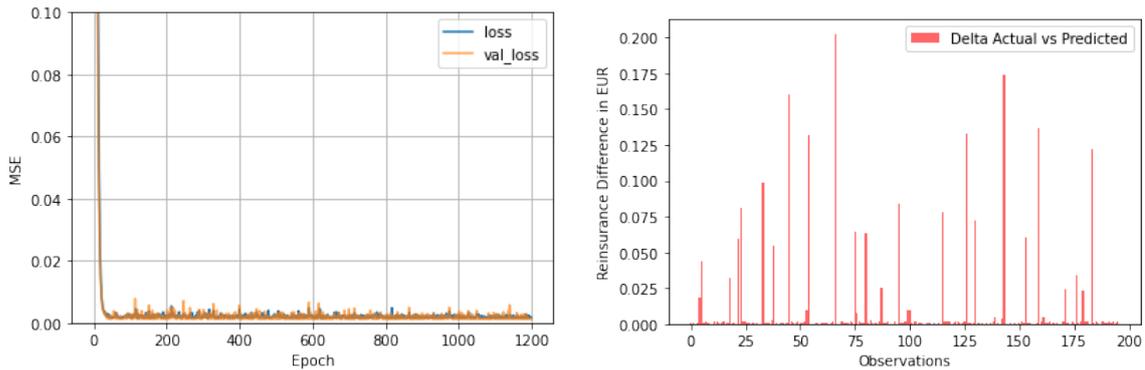


Abbildung 5.8: ReLU Modell 2 mit 1200 Epochen

Zum Schluss dieses Abschnittes werden alle Ergebnisse nochmals übersichtlich zusammengefasst:

Tabelle 5.3: Ergebnisse der Modelle mit Validierungsdaten

Modell	Layer	Knoten	Epochen	Lernrate	MAE	MSE
Tanh 1	3	192	800	0.001	0.0325	0.0027
Tanh 2	4	512	800	0.0001	0.0173	0.0017
Tanh 3	4	512	1200	0.0001	0.0137	0.0012
ReLU 1	3	192	800	0.001	0.0131	0.0018
ReLU 2	4	576	800	0.0001	0.0120	0.0012
ReLU 3	4	576	1200	0.0001	0.0110	0.0011

5.2.3 Vorhersage

In diesem Abschnitt werden die Daten der „AVÖ 2018-P - Rechnungsgrundlagen für Pensionsversicherung“ [KHS18] (kurz: „AVÖ 2018-P“) herangezogen und Vorhersagen mithilfe des vortrainierten Neuronalen Netzwerks gemacht. Das Netzwerk wurde mit Daten trainiert, die stochastische Komponenten enthalten, welche das Langlebkeitsrisiko widerspiegeln. Somit sollen die Rückversicherungsbeiträge unter Berücksichtigung des Langlebkeits-

keitsrisikos vorhergesagt werden. Es gelten dieselben Annahmen wie in Abschnitt 5.1.

Aus den Daten der „AVÖ 2018-P“ werden die Überlebenswahrscheinlichkeiten der Angestellten für Männer und Frauen ab dem Alter von 60 Jahren herangezogen. Diese dienen nur wissenschaftlichen Zwecken und dürfen nicht explizit angegeben werden. Die Wahrscheinlichkeiten, das zugehörige Alter und Geschlecht bilden den Input für das Deep Learning Modell. Anhand dieser Daten wird die Höhe der Rückversicherung vorhergesagt.

Die Ergebnisse der Evaluierung mittels den Daten der „AVÖ 2018-P“ aller Modelle aus den Abschnitten 5.2.1 und 5.2.2 wird zusammengefasst:

Tabelle 5.4: Ergebnisse der Modelle mit Daten der „AVÖ 2018-P“

Modell	Layer	Knoten	Epochen	Lernrate	MAE	MSE
Tanh 1	3	192	800	0.001	0.0515	0.0102
Tanh 2	4	512	800	0.0001	0.0310	0.0072
Tanh 3	4	512	1200	0.0001	0.0260	0.0055
ReLU 1	3	192	800	0.001	0.0354	0.0099
ReLU 2	4	576	800	0.0001	0.0308	0.0077
ReLU 3	4	576	1200	0.0001	0.0240	0.0055

Allgemein sind MAE und MSE für diese Daten etwas schlechter als für die Validierungsdaten. Dies ist jedoch nicht überraschend, da Simulationen echte Daten nie exakt darstellen können. Da der Unterschied zwischen MAEs und MSEs der Validierungsdaten und der Daten der „AVÖ 2018-P“ sehr gering ist handelt es sich trotzdem um ein gutes Modell.

Sowohl bei den Tanh Modellen als auch bei den ReLU Modellen schneidet jeweils das Modell 3 für die Daten der „AVÖ 2018-P“ am besten ab. Allerdings kann man trotzdem sagen, dass alle Modelle auch mit den Daten der „AVÖ 2018-P“ relativ gut abschneiden, da die Fehler trotzdem eine akzeptable Größe aufweisen.

Die insgesamt besten Ergebnisse liefert das ReLU Modell 3. Somit wurde ein Deep Learning Modell auf simulierten Daten erstellt, welches ebenso für reale Daten sehr gute Vorhersagen für den Betrag der Rückversicherung mit verschiedensten Annahmen an die Sterblichkeit macht.

Das Modell kann sicherlich weiter optimiert werden, da es momentan noch an gewisse Annahmen gebunden ist. Somit könnte diese Herangehensweise, das Langlebkeitsrisiko mittels Deep Learning abzusichern, in Zukunft immer mehr an Bedeutung gewinnen. Grund dafür ist, dass die vorgestellten Algorithmen sehr genaue Ergebnisse liefern, sobald die richtigen Hyperparameter gefunden werden. Das Keras Package bietet einfache Implementierungen Neuronaler Netzwerke an, welche schon nach kurzen Trainingsphasen zuversichtliche Ergebnisse liefern.

Abbildungsverzeichnis

2.1	Extrapolation of the mortality profile	5
2.2	Eine XL-Rückversicherungsvereinbarung	10
2.3	Eine Stop-Loss Rückversicherungsvereinbarung	10
3.1	(Summen-)Exzedentenrückversicherung	17
3.2	Formen der proportionalen Rückversicherung	18
3.3	Formen der nicht-proportionalen Rückversicherung	21
4.1	Aufbau eines Perzeptrons mit Aktivierungsfunktion ϕ und Bias b	33
4.2	Feedforward Neuronales Netzwerk mit zwei hidden Layern	35
4.3	Aktivierungsfunktionen	38
4.4	Abfallende Lernrate ohne Momentum	43
4.5	Abfallende Lernrate mit Momentum	44
5.1	Simulationen Männer mit $\mu = 0.00069$, $\sigma = 0.0001$ und $\gamma = 0.108$	49
5.2	Simulationen Frauen mit $\mu = 0.000418$, $\sigma = 0.00009$ und $\gamma = 0.1211$	50
5.3	Tanh Modell mit 3 hidden Layern mit jeweils 64 Knoten und 800 Epochen	55
5.4	Tanh Modell mit 4 hidden Layern, 512 (1×256 , 1×128 , 2×64) Knoten und 800 Epochen	55
5.5	Tanh Modell 2 mit 1200 Epochen	56
5.6	ReLU Modell mit 3 hidden Layern mit jeweils 64 Knoten und 800 Epochen	57
5.7	ReLU Modell mit 4 hidden Layern, 576 (1×256 , 2×128 , 1×64) Knoten und 800 Epochen	57
5.8	ReLU Modell 2 mit 1200 Epochen	58

Literaturverzeichnis

- [A⁺18] Charu C Aggarwal et al. Neural networks and deep learning. *Springer*, 2018.
- [Bau19] Maria-Theresa Baumgarten. Neuronale Netze zur SCR Berechnung. Masterarbeit, Technische Universität Wien, 2019.
- [BGTW19] Hans Buehler, Lukas Gonon, Josef Teichmann, and Ben Wood. Deep hedging. *Quantitative Finance*, 19(8):1271–1291, 2019.
- [Car83] R.L. Carter. *Reinsurance*. Springer Netherlands, 2nd edition, 1983.
- [CFvBM20] Eamon Comerford, Paul Fulcher, Rik van Beers, and Rosemary Maher. *Reinsurance as a capital management tool for life insurers*. Milliman Research Report, 2020.
- [GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [Ger19] Caroline Gerharter. Deep Learning in Life Insurance Risk Prediction. Masterarbeit, Technische Universität Wien, 2019.
- [Hig01] Desmond J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 43(3):525–546, 2001.
- [Hor91] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.
- [KHS18] Reinhold Kainhofer, Jonas Hirz, and Alexander Schubert. AVÖ 2018-P: Rechnungsgrundlagen für die Pensionversicherung. Dokumentation, 2018.
- [Koc18] Marlene Koch. Rückversicherung von Lebensrisiken. Masterarbeit, Technische Universität Wien, 2018.
- [LC92] Ronald D Lee and Lawrence R Carter. Modeling and forecasting us mortality. *Journal of the American statistical association*, 87(419):659–671, 1992.
- [Led21] Johannes Lederer. Activation functions in artificial neural networks: A systematic overview. *arXiv preprint arXiv:2101.09957*, 2021.
- [Lee00] Ronald Lee. The lee-carter method for forecasting mortality, with various extensions and applications. *North American actuarial journal*, 4(1):80–91, 2000.

- [LM01] Ronald Lee and Timothy Miller. Evaluating the performance of the lee-carter method for forecasting mortality. *Demography*, 38(4):537–549, 2001.
- [Mit97] T.M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997.
- [Nor97] Fredrik Norström. The gompertz-makeham distribution, 1997.
- [P+02] Ermanno Pitacco et al. Longevity risk in living benefits. In *third annual*. Citeseer, 2002.
- [Pfe86] Christoph Pfeiffer. *Einführung in die Rückversicherung*. Gabler, 3rd edition, 1986.
- [PG17] Josh Patterson and Adam Gibson. *Deep learning: A practitioner's approach*. O'Reilly Media, Inc., 2017.
- [Rhe19] Thorsten Rheinländer. Höhere Lebensversicherungsmathematik. Vorlesungsmitschrift, Technische Universität Wien, 2019.
- [RHT05] Petra Riemer-Hommel and Thomas Trauth. The challenge of managing longevity risk. In *Risk Management*, pages 391–406. Springer, 2005.
- [Rud16] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [SSA17] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *towards data science*, 6(12):310–316, 2017.
- [SSK09] Prof. W. Schachermayer, Prof. U. Schmock, and Prof. N. Kusolitsch. Lebensversicherungsmathematik. Skriptum, Technische Universität Wien, 2009.
- [ZG08] Alexander Zwirchmayr and Peter Grandits. Sachversicherungsmathematik. Vorlesungsmitschrift, Technische Universität Wien, 2008.