

# **Automatisierung von Geschäftsprozessen durch künstliche Intelligenz: Eine explorative Untersuchung des Potentials am Beispiel der Rechnungseinreichung in der Krankenversicherung**

Bachelor Thesis

Zürcher Fachhochschule

**HWZ Hochschule für Wirtschaft Zürich**

Eingereicht bei

Dr. Oliver Zenklusen

vorgelegt von: Sven Tschui

Matrikelnummer: 15-522-345

Studiengang: Bachelor of Science Wirtschaftsinformatik

Ort, Datum Zürich, 30. April 2019



# Abstract

Die künstliche Intelligenz prägt viele Hollywood Kassenschlager. Diese stellen die künstliche Intelligenz oft als Wesen mit menschlichen Zügen, die diesen überlegen sind, dar. Doch bezeichnet der Begriff künstliche Intelligenz Technologien, mit welchen Computern beigebracht wird, flexible und rationale Entscheidungen zu treffen. Es werden dem Computer dabei keine Regeln, sondern Eingabe- und erwartete Ausgabewerte vorgegeben. Der Computer soll anhand dieser Beispiele lernen, für neue Eingabewerte eine Antwort vorherzusagen.

Heutige Anwendungen der künstlichen Intelligenz sind spezifisch für die zu lösende Aufgabe. Diese sogenannte aufgabenspezifische künstliche Intelligenz, die geschaffen wurde um die Aufgabe A zu lösen, ist im Gegensatz zu uns Menschen oder einer ganzheitlichen künstlichen Intelligenz, nicht geeignet die Aufgabe B zu lösen.

Das Forschungsgebiet der künstlichen Intelligenz ist trotz seines Alters immer wieder von neuen Entwicklungen geprägt. Zur Zeit werden die Technologien aus diesem Gebiet meist nur von Technologie-Giganten und Start-Ups verwendet. Durch Initiativen wie [www.fast.ai](http://www.fast.ai) wird das Forschungsgebiet zugänglicher und es wird für die breiten Massen immer einfacher, Systeme mit künstlicher Intelligenz zu schaffen.

Aufgrund dieser Entwicklungen setzt sich diese Arbeit zum Ziel, herauszufinden, ob Geschäftsprozesse in kleineren Unternehmen durch eigenentwickelte künstliche Intelligenz automatisiert werden können.

Zur Untersuchung werden relevante Grundlagen des Forschungsgebietes der künstlichen Intelligenz erläutert. Darauf aufbauend wird in einem explorativen Teil ein Fallbeispiel erarbeitet. Dabei wird die Automatisierung des Prozesses zur Rechnungseinreichung bei der AXA Gesundheitsvorsorge mit Hilfe von künstlicher Intelligenz analysiert. Zur Analyse zweier Aspekte der Automatisierung des Prozesses werden zwei Modelle zur Klassifizierung der Rechnungen und eines zur Informationsextraktion implementiert respektive optimiert.

Zur Klassifizierung der Rechnungen wurde ein Bild-basiertes sowie ein Text-basiertes Modell implementiert und evaluiert. Das Text-basierte Modell erzielte dabei bessere Resultate. Es erreicht eine Trefferquote von 98.4%. Im Rahmen der Fehleranalyse wurde Optimierungspotential im Bereich der OCR Erkennung und des Word embeddings aufgezeigt, durch welches die Trefferquote in Zukunft noch weiter erhöht werden kann.

Zur Informationsextraktion aus den Rechnungen wurde ein Bild-basiertes Modell evaluiert. Das Modell erreicht auf einem Datensatz von 900 Rechnungen von Optikern gute, aber dennoch keine zufriedenstellende Resultate. Das Modell wurde erweitert, damit es alle relevanten Informationen zur Automatisierung einer Rechnung liefern kann. Des weiteren wurde das Modell auf Rechnungen eines Typs spezialisiert. Das Modell wurde auf Rechnungen von Fielmann trainiert, wobei es eine Mean Average Precision (mAP@0.5) von 94% erreicht. Das Modell wurde ausserdem auf Rechnungen von Visilab trainiert. Das Modell erzielt hier aufgrund der geringen Zahl an Trainingsdaten eine etwas tiefere Mean Average Precision. Diese guten Ergebnisse können durch das in der Arbeit aufgezeigte Optimierungspotential noch weiter verbessert werden.

Die Ergebnisse und das aufgezeigte Optimierungspotential belegen, dass eine Automatisierung des Prozesses möglich ist. Die Modelle müssen vor produktivem Einsatz weiter optimiert werden. Der Aufwand für die Optimierung darf nicht unterschätzt werden, denn das Modell zur Informationsextraktion muss spezifisch für jeden Leistungserbringer trainiert werden. Der AXA Gesundheitsvorsorge wird empfohlen, weiter in die Automatisierung mit Hilfe von künstlicher Intelligenz zu investieren und einen inkrementellen Rollout eines Systems mit künstlicher Intelligenz anzustreben.

Initiativen wie das DAWN Projekt, welche die künstliche Intelligenz immer zugänglicher machen, und die Ergebnisse aus dem Fallbeispiel zeigen, dass die künstliche Intelligenz auch von kleineren Unternehmen mit limitiertem Investitionskapital verwendet werden kann, um Geschäftsprozesse zu automatisieren. Trotz der noch immer hohen Investitionskosten ist es auch für kleinere Unternehmen ratsam, in die künstliche Intelligenz zu investieren, damit sie sich einen Wettbewerbsvorteil schaffen beziehungsweise einen solchen halten können.





# Inhaltsverzeichnis

<b>Ehrenwörtliche Erklärung</b>	<b>IX</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Zielsetzung . . . . .	3
1.2 Vorgehen . . . . .	4
1.3 Inhaltliche Abgrenzung . . . . .	4
<b>2 Automatisierung eines Geschäftsprozesses</b>	<b>5</b>
2.1 Gründe zur Automatisierung eines Geschäftsprozesses . . . . .	5
2.2 Risiken durch die Automatisierung . . . . .	7
2.3 Automatisierung durch Anwendung künstlicher Intelligenz . . . . .	7
2.4 Anwendungsbeispiele künstlicher Intelligenz . . . . .	8
2.4.1 Ping An Insurance Co. of China Ltd. . . . .	9
2.4.2 Blue River Technology . . . . .	9
2.4.3 Infervision . . . . .	9
<b>3 Künstliche Intelligenz: Grundlagen und Ansätze in der Praxis</b>	<b>11</b>
3.1 Neuronale Netzwerke . . . . .	11
3.2 Tiefe neuronale Netzwerke . . . . .	14
3.3 Backpropagation . . . . .	17
3.4 Over- und Underfitting . . . . .	17
3.5 Convolutional Neural Network . . . . .	19
3.6 Long-Short-Term-Memory Netzwerke . . . . .	23
3.7 Texterkennung . . . . .	24
3.8 Word embedding . . . . .	25
3.9 Korrektur von Rechtschreibung und Grammatik . . . . .	27
3.10 Natural Language Processing . . . . .	29
3.11 Informationsextraktion aus natürlichen Texten . . . . .	29
3.12 Klassifizierung von Bildern . . . . .	31
3.13 Objekterkennung in Bildern . . . . .	32
3.14 Transfer Learning . . . . .	33

3.15	Messkriterien zur Bewertung eines Systems mit künstlicher Intelligenz . . . . .	35
3.15.1	Trefferquote . . . . .	35
3.15.2	Genaugkeit . . . . .	36
3.15.3	Sensitivität . . . . .	36
3.15.4	F-Mass . . . . .	37
3.15.5	Loss und Loss-Funktion . . . . .	38
3.15.6	Intersection over Union . . . . .	38
3.15.7	Average Precision . . . . .	40
3.15.8	Mean Average Precision . . . . .	43
3.16	Fehleranalyse . . . . .	43
3.17	Design eines Systems mit künstlicher Intelligenz . . . . .	44
3.17.1	Design eines neuronalen Netzwerkes . . . . .	45
3.18	End-to-end Entwicklung von künstlicher Intelligenz . . . . .	46
<b>4</b>	<b>Automatisierung der Rechnungseinreichung der AXA Gesundheitsvorsorge</b>	<b>47</b>
4.1	Einführung in das Fallbeispiel . . . . .	47
4.1.1	Aktueller Prozess der Rechnungseinreichung . . . . .	50
4.2	Anforderungen . . . . .	52
4.3	Vorgehen und Methodik . . . . .	54
4.4	Teil 1 - Klassifizierung von Rechnungen . . . . .	54
4.4.1	Bild-basierte Rechnungsklassifizierung . . . . .	55
4.4.2	Text-basierte Rechnungsklassifizierung . . . . .	59
4.4.3	Schlussfolgerungen . . . . .	66
4.5	Teil 2 - Informationsextraktion . . . . .	66
4.5.1	Bild-basierte Informationsextraktion . . . . .	67
4.5.2	Bild-basierte Informationsextraktion pro Rechnungstyp . . . . .	71
4.5.3	Vervollständigung des präsentierten Ansatzes . . . . .	76
4.5.4	Ausblick . . . . .	77
4.5.5	Schlussfolgerungen . . . . .	77
<b>5</b>	<b>Empfehlungen und Schlussfolgerungen</b>	<b>79</b>
5.1	Empfehlungen an die AXA Gesundheitsvorsorge . . . . .	79
5.2	Schlussfolgerungen . . . . .	80
<b>6</b>	<b>Ausblick</b>	<b>83</b>
<b>7</b>	<b>Kritische Reflexion</b>	<b>85</b>
<b>8</b>	<b>Anhang</b>	<b>87</b>
8.1	Literaturverzeichnis . . . . .	87
8.2	Tabellen- und Abbildungsverzeichnis . . . . .	93
8.3	Sourcecode . . . . .	95





# Ehrenwörtliche Erklärung

Ich bestätige hiermit, dass ich

- die vorliegende Thesis selbständig und ohne Benützung anderer als der angegebenen Quellen und Hilfsmittel anfertigte,
- die benutzten Quellen wörtlich oder inhaltlich als solche kenntlich machte,
- diese Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungskommission vorlegte.

Zürich, 30. April 2019

.....  
Sven Tschui



# 1 | Einleitung

Science-Fiction Kassenschlager aus Hollywood zeigen künstliche Intelligenz als Wesen mit menschlichen Zügen, die diesen oft überlegen sind. Solche Darstellungen schüren Ängste. Einige dieser Ängste, wie die Ausrottung der Menschheit, scheinen in weiter Ferne. Andere, am Arbeitsplatz durch einen Roboter abgelöst zu werden, scheinen realer als je zuvor (Lu, Li, Chen, Kim & Serikawa, 2018).

Die Thematik wird nicht nur von Massenmedien aufgegriffen, sondern auch in wissenschaftlichen Artikeln behandelt. So schreibt Tredinnick (2017), Forscher in den Gebieten der digitalen Kultur, Technologien und neuen Medien, dass das Jahr 2017 das Jahr zu sein verspricht, in welchem die künstliche Intelligenz aus Film und Fiktion in die Arbeitswelt übergeht.

Bereits 1951 sagte Turing, wir hätten erwarten sollen, dass die Maschinen die Kontrolle übernehmen werden. Doch so weit sind wir noch nicht. Dennoch bezeichnet Tredinnick (2017) die künstliche Intelligenz als die vierte industrielle Revolution mit einschneidenden Veränderungen in unserer Wirtschaft und unserem Leben.

Doch was genau ist künstliche Intelligenz? Künstliche Intelligenz bezeichnet Technologien, mit welchen Computern beigebracht wird, flexible und rationale Entscheidungen zu treffen. Im Gegensatz zur klassischen Vorgehensweise werden dem Computer keine klaren Regeln vorgegeben, anhand welcher er operieren soll, sondern er wird mit Eingabewerten und erwarteten Ausgaben trainiert, selbst einen Lösungsweg zu finden (Tredinnick, 2017).

Die Geschichte der künstlichen Intelligenz hat früh begonnen. Weizenbaum entwickelte bereits 1966 einen Chatbot, der durch einfache Techniken in der Lage war, den Eindruck zu vermitteln, mit einem intelligenten System zu kommunizieren. Obwohl das System auf einfache Regeln zurückgriff, um eine passende und sehr offene Aussage zu formulieren, markierte es den Startschuss für die Forschung im Gebiet der Verarbeitung von natürlicher Sprache (Tredinnick, 2017).

1997 gelang IBM mit Deep Blue der erste Sieg in Schach gegen den damaligen Weltmeister Garry Kasparow (Campbell, Hoane und Hsu, 2002 in Tredinnick, 2017). 2011 gelang es IBM schliesslich mit ihrem System namens Watson<sup>1</sup> die amerikanische Quiz-Show Jeopardy zu gewinnen. Dieser Sieg ist ein Meilenstein für die künstliche Intelligenz. Mit Watson leistete IBM Pionierarbeit bei der Sammlung und Verarbeitung von unstrukturierten sowie strukturierten Daten (Tredinnick, 2017).

Die aufgeführten Beispiele sind Ausprägungen der sogenannten schwachen oder aufgabenspezifischen künstlichen Intelligenz. Diese Art der künstlichen Intelligenz ist auf eine spezifische Aufgabe ausgerichtet und kann diese oft besser ausführen als ein Mensch. Im Gegensatz zu einem Menschen ist diese Art der künstlichen Intelligenz aber nicht fähig andere Aufgaben zu lösen (Lu et al., 2018).

Künstliche Intelligenz, welche mit gleicher Flexibilität und Kreativität wie Menschen auf Ihre Umwelt reagiert, wird generelle künstliche Intelligenz genannt. Im Jahre 1993 sagte Vinge, dass die technologische Singularität<sup>2</sup>, welche durch generelle künstliche Intelligenz erreicht würde, nur noch 30 Jahre entfernt sei. Diese 30 Jahre sind bald erreicht, doch die generelle künstliche Intelligenz steht noch immer in weiter Ferne. Statt einer generellen künstlichen Intelligenz näher zu kommen, wird immer mehr erkannt, wie komplex eine solche ist (Tredinnick, 2017).

Die Zeit der generellen künstlichen Intelligenz ist noch nicht gekommen, doch findet die aufgabenspezifische künstliche Intelligenz bereits heute Anwendung (Tredinnick, 2017). Im Bereich der Produktion von Saatgut gibt es bereits mehrere Studien, welche die Lösung der Problematiken der Krankheitserkennung, Saatgutqualität sowie Phänotypisierung unter Anwendung von computergestützter Bildverarbeitung mit künstlicher Intelligenz behandeln (Patrício & Rieder, 2018).

Auch um die Produktion des neuen Airbus A350 schnellstmöglich auf Hochtouren zu bringen, wurde künstliche Intelligenz angewendet. Ein System, welches von Airbus entwickelt wurde, ermöglicht, dank künstlicher Intelligenz, in 70% aller Unterbrüche der Produktion in kürzester Zeit eine Lösung für die Wiederinbetriebnahme auszuarbeiten (Ransbotham, Kiron, Gerbert & Reeves, 2017).

---

<sup>1</sup>IBM Watson ist ein System, welches durch Techniken aus den Gebieten des Natural Language Processing und des Machine Learning sowie durch das Mining von strukturierten und unstrukturierten Daten in der Lage ist, Fragen zu beantworten (Tredinnick, 2017).

<sup>2</sup>Als technologische Singularität wird jener Zeitpunkt bezeichnet, zu welchem die künstliche Intelligenz die menschliche Intelligenz überholt. Ab diesem Zeitpunkt wird sich die Technologie rasend schnell selbst weiterentwickeln. Gemäss den Vertretern dieser Theorie stellt die technologische Singularität eine enorme Gefahr für die Menschheit dar (Tredinnick, 2017).

## 1.1 Zielsetzung

---

Ping An Insurance Co. of China Ltd., eine der grössten Versicherungsgesellschaften von China, verwendet künstliche Intelligenz zur Automatisierung von diversen Prozessen. Die Versicherung beschäftigt 110 Data Scientists, die bereits etliche Initiativen im Bereich der künstlichen Intelligenz umgesetzt haben (Ransbotham et al., 2017).

Neben diesen Pionieren erwähnen Ransbotham et al. (2017) in Ihrer Untersuchung aber auch, dass nur 14% der Befragten denken, dass künstliche Intelligenz aktuell einen hohen Einfluss auf Ihre Angebote und Dienstleistungen hat. Jedoch denken 63%, dass sich dies in den nächsten 5 Jahren ändern wird und die künstliche Intelligenz einen entscheidenden Wettbewerbsvorteil bieten kann (Ransbotham et al., 2017).

Auch im The Economist (2018) wird der mögliche Wettbewerbsvorteil durch die Anwendung von künstlicher Intelligenz angesprochen. Ausserhalb des Technologie-Sektors, in Branchen, welche aktuell durch den Konkurrenzkampf geprägt sind, werden grosse Firmen durch die Anwendung künstlicher Intelligenz noch grösser werden und sich zu Monopolen entwickeln.

In den letzten Monaten wurde die künstliche Intelligenz immer greifbarer. Initiativen wie die Webseite [www.fast.ai](http://www.fast.ai), welche zum Ziel haben die Programmierung künstlicher Intelligenz der breiten Masse zugänglich zu machen, ermöglichen es, immer mehr Anwendungsfälle umzusetzen.

In dieser Arbeit wird diskutiert, ob und wieweit Anwendungen der künstlichen Intelligenz für kleinere Unternehmen mit limitiertem Budget und Ressourcen möglich sind. Die Diskussion orientiert sich an folgender Forschungsfrage:

Können Geschäftsprozesse in kleineren Unternehmen durch eigenentwickelte künstliche Intelligenz automatisiert werden?

### 1.1 Zielsetzung

Diese Arbeit hat zum Ziel, herauszufinden, ob Anwendungen der künstlichen Intelligenz in kleineren Unternehmen möglich sind. Um sich der Forschungsfrage anzunähern, wird ein Überblick über die für diese Arbeit relevanten Begriffe, Techniken und Konzepte der künstlichen Intelligenz gegeben.

Die Forschungsfrage wird an einem konkreten Fallbeispiel, der Rechnungseinreichung bei der AXA Gesundheitsvorsorge, untersucht. Im Rahmen dieser Arbeit wird ein Prototyp entwickelt, welcher die Machbarkeit der Automatisierung der Rechnungseinreichung aufzeigt. Ein weiteres Ziel dieser Arbeit ist es, konkrete Empfehlungen an die AXA Gesundheitsvorsorge zu geben, wie im vorliegenden Fallbeispiel weiter verfahren werden soll.

## 1.2 Vorgehen

Im Kapitel 2 wird dargelegt, weshalb eine Automatisierung eines Geschäftsprozesses erstrebenswert sein kann. Neben den Gründen für eine Automatisierung werden auch die Risiken durch eine Automatisierung erläutert. Es wird dargelegt, warum künstliche Intelligenz benötigt wird, um Geschäftsprozesse zu automatisieren. Es werden ausserdem Anwendungsbeispiele künstlicher Intelligenz in der Automatisierung aufgezeigt.

Im Kapitel 3 werden Themengebiete, Konzepte und Techniken im Zusammenhang mit der künstlichen Intelligenz erläutert. Ein wichtiger Bestandteil dabei ist die Funktionsweise und Anwendung von neuronalen Netzen sowie Konzepte aus dem Natural Language Processing. Die Erklärungen dieses Kapitels dienen als Grundlage für den explorativen Teil dieser Arbeit, welcher im Kapitel 4 erläutert wird.

Im Kapitel 4 wird in einem explorativen Vorgehen das Fallbeispiel der AXA Gesundheitsvorsorge erarbeitet. Zu Beginn wird das Fallbeispiel selbst sowie der Prozess der Rechnungseinreichung bei der AXA Gesundheitsvorsorge und die damit verbundene Problemstellung erläutert. Folgend werden Anforderungen an die Automatisierung von Rechnungen von Optikern und Fitnesscentern definiert. Es werden zwei Teilespekte der Problemstellung durch je zwei Experimente diskutiert. Der erste Teil widmet sich der Klassifizierung der Rechnungen. Im zweiten Teil werden Lösungen zur Extraktion von Informationen aus den Rechnungen vorgestellt. In beiden Teilen werden jeweils die Ergebnisse diskutiert und Fehlerquellen analysiert.

Im Kapitel 5 wird die Forschungsfrage diskutiert und es werden Schlussfolgerungen aufgrund der Ergebnisse der Experimente getroffen. Es wird ausserdem eine Empfehlung an die AXA Gesundheitsvorsorge zum weiteren Vorgehen abgegeben.

Kapitel 6 gibt einen Ausblick zu künftigen Forschungsfeldern sowie Techniken, welche für das erarbeitete Fallbeispiel relevant sind.

Die Arbeit wird mit einer kritischen Reflexion im Kapitel 7 abgeschlossen.

## 1.3 Inhaltliche Abgrenzung

Als Produkt dieser Arbeit entsteht ein Prototyp, der zum Zweck hat, die Forschungsfrage zu beantworten. Der Prototyp soll als Grundlage zur Entwicklung eines produktionsreifen Systems dienen, hat selbst aber keinerlei Anspruch produktionsreif zu sein.

Techniken aus dem Bereich der künstlichen Intelligenz und des Machine Learnings werden nur oberflächlich erläutert, um ein Verständnis zu gewährleisten. Eine vollumfängliche Einführung in das Themengebiet der künstlichen Intelligenz ist nicht Teil dieser Arbeit.

## 2 | Automatisierung eines Geschäftsprozesses

In diesem Kapitel wird erläutert, weshalb es für ein Unternehmen sinnvoll ist, einen Geschäftsprozess zu automatisieren und welche Risiken damit einhergehen. Es wird beschrieben, warum die künstliche Intelligenz bei der Automatisierung eines Geschäftsprozesses zur Anwendung kommt. Weiter werden Beispiele der Anwendung von künstlicher Intelligenz zur Automatisierung vorgestellt.

### 2.1 Gründe zur Automatisierung eines Geschäftsprozesses

„Überdurchschnittliche unternehmerische Leistung beruhen langfristig auf Wettbewerbsvorteilen, mit welchen sich ein Unternehmen behaupten kann“ (Capaul & Steingruber, 2010, S. 104). Zur Erreichung eines solchen Wettbewerbsvorteil bedient sich ein Unternehmen an einer Wettbewerbsstrategie. Porter strukturiert diese Strategien nach dem strategischen Vorteil und dem strategischen Zielobjekt (vgl. Abbildung 1) (Capaul & Steingruber, 2010).

Abbildung 1: Wettbewerbsstrategien nach Porter, systematisiert nach strategischem Vorteil und strategischem Zielobjekt

Strategisches Zielobjekt	Strategischer Vorteil (Leistung oder Kosten)		
	Branchenweit (Gesamtmarktabdeckung)	Differenzierung (Qualitätsführerschaft)	Kostenführerschaft
	Beschränkung auf Segment (Teilmarktabdeckung)		Konzentration auf Nischen

Quelle: Capaul und Steingruber (2010)

Als strategische Vorteile sieht Porter eine bessere Leistung oder tiefere Kosten als Konkurrenten. Möchte ein Unternehmen langfristig Erfolg haben, so muss es sich entweder über die Leistung oder die Kosten abheben (Capaul & Steingruber, 2010). Die Automatisierung eines Geschäftsprozesses kann bei der Erreichung beider dieser Vorteile helfen und ist deshalb für ein Unternehmen erstrebenswert.

Um eine Kostenführerschaft zu erzielen, sind tiefe Selbstkosten sehr wichtig (Capaul & Steingruber, 2010). Eine Automatisierung kann hohe Selbstkosten, beispielsweise durch hohen manuellen Aufwand, reduzieren und somit eine Kostenführerschaft ermöglichen.

Eine Differenzierung im Bereich der Leistung kann durch eine Automatisierung gleich auf zwei verschiedene Arten unterstützt werden. Automatisierte Prozesse weisen weniger Fehler auf, Produkte oder Dienstleistungen können also in einer besseren Qualität angeboten werden. Kergaßner (2012) spricht in der IT Administration von einer Fehlerquote von 10%, selbst bei einfachen, sich wiederholenden Tätigkeiten, wenn diese manuell ausgeführt werden. Werden diese Tätigkeiten automatisiert, reduziert sich die Fehlerquote. Ähnliche Beobachtungen wurden auch von Uettwiler-Geiger (2005) im Bereich von medizinischen Laboren gemacht. In diesem Bereich konnte die Fehlerquote durch die Automatisierung stark reduziert werden. Neben der Reduktion der Fehlerquote kann die Automatisierung eines Geschäftsprozesses auch einen Zusatznutzen für Kundinnen und Kunden bedeuten. Im Beispiel einer Krankenkasse könnte eine automatisierte Leistungsabwicklung die Durchlaufzeit bis zur Auszahlung einer eingereichten Rechnung verkürzen. Gesundheitskosten können zu einer finanziellen Notlage führen, daher ist es Kundinnen und Kunden wichtig, das ihnen zustehende Geld schnell ausbezahlt zu bekommen.

Werden zeitgleich beide strategischen Vorteile erzielt, spricht man von einer hybriden Strategie. Eine solche hybride Strategie bietet den höchsten Return on Investment und kann einem Unternehmen zu einer Quasi-Monopolstellung verhelfen (Lombriser & Abplanalp, 2015). Gute Beispiele für eine Monopolstellung durch die Nutzung von modernen Technologien zur automatisierten Bereitstellung von Produkten und Dienstleistungen sind Technologie-Giganten wie Amazon. Amazon ermöglicht dank einem hohen Grad an Automatisierung ein Kundenerlebnis wie dies kein anderen Online-Händler zuvor erreichen konnte. Trotz der Zusatznutzen die Amazon verspricht, bleiben die Preise tief. Dies ist nur durch einen hohen Automatisierungsgrad möglich (Kha, 2000).

## 2.2 Risiken durch die Automatisierung

Die Automatisierung von Geschäftsprozessen ist erstrebenswert, geht aber nicht ohne Risiken einher. Nach der erfolgreichen Einführung eines Systems zur Automatisierung, wird oft erst im Laufe der Zeit erkannt, dass nicht nur positive Effekte erzielt wurden. Mit der Einführung jeder neuen Maschine oder Software entsteht neues Potential für Probleme und Fehler. Diese werden meist durch fehlende Kommunikation zwischen dem Anwender und dem System verursacht - der Anwender ist oft überrascht über das Verhalten des Systems. Diese Problematik wird bei der Konzeption eines Systems verursacht. Aus diesem Grund ist es wichtig, bereits beim Design eines Systems darauf zu achten, wie ein Anwender damit umgeht (Sarter, Woods & Billings, 1997).

Ein neues System wird meist nicht ohne Fehler eingeführt. Beispielsweise hat das Betriebssystem aus dem Hause Microsoft, eine traditionelle, regelbasierte Software, zum Zeitpunkt der Veröffentlichung einen Fehler pro 2'000 Zeilen Code. Das bedeutet, dass Windows XP zum Zeitpunkt der Veröffentlichung, mit über 40 Millionen Zeilen Code, mindestens 20'000 Fehler hatte (The Economist, 2010).

Während bei traditioneller Software Fehler auf einzelne Regeln beziehungsweise eine Zeile Code zurückverfolgt werden können, ist dies bei komplexeren Systemen, wie beispielsweise neuronalen Netzwerken, nicht möglich. Fehler beziehungsweise Ungenauigkeiten sind in einem solchen System ein fester Bestandteil. Anstelle diese ganzheitlich zu beseitigen, wird versucht diese zu minimieren(van Rijsbergen, 1979).

## 2.3 Automatisierung durch Anwendung künstlicher Intelligenz

Das Polanyi Paradox besagt, dass wir Menschen mehr Wissen, als wir beschreiben oder erklären können. Diese Problematik gilt auch für klassische Computeranwendungen. Wir waren über lange Zeit nicht in der Lage, dem Computer Dinge beizubringen, die wir nicht erklären konnten. Da Computer bisher immer anhand von Menschen definierter Regeln operierten, waren sie bisher limitiert in den Dingen, die sie erledigen konnten (Brynjolfsson & McAfee, 2017).

Die künstliche Intelligenz bietet die Möglichkeit, diese Limitierung zu Umgehen. Anstelle Software anhand vordefinierter Regeln zu schreiben, lernt die Software die Regeln selbst. So kann die Software lernen, was wir Menschen nicht ausdrücken können (Brynjolfsson & McAfee, 2017).

Künstliche Intelligenz wird als die wichtigste Allzweck-Technologie unserer Zeit gehandelt. Vergleiche mit der Dampfkraft, Elektrizität und dem Verbrennungsmotor liegen sehr nahe (Brynjolfsson & McAfee, 2017).

Die künstliche Intelligenz findet in unterschiedlichen Bereichen Anwendung. Obwohl diese Anwendungen noch nicht perfekt sind, kann die Leistung eines Menschen bereits übertrffen werden. Ein gutes Beispiel dafür ist die Erkennung von Objekten auf Bildern. Bereits im Jahr 2015 konnte die menschliche Fehlerquote von ca. 5% von einer künstlichen Intelligenz unterboten werden<sup>3</sup>. Diese vielversprechenden Ergebnisse zeigen, warum die künstliche Intelligenz für die Automatisierung von Geschäftsprozessen eine immer wichtigere Rolle spielt (Brynjolfsson & McAfee, 2017).

Eine Studie von McKinsey belegt, dass die künstliche Intelligenz einem Unternehmen das Potential zur Disruption geben kann. Unternehmen, welche die künstliche Intelligenz früh einführen und mit einer proaktiven Strategie kombinieren, haben einen höheren Gewinn als die Konkurrenz (McKinsey Global Institute, 2017).

## 2.4 Anwendungsbeispiele künstlicher Intelligenz

Die Studie des McKinsey Global Institute (2017) zeigt, dass die grossen Investitionen in die künstliche Intelligenz noch immer von den Technologie-Giganten und Digital Native Unternehmen wie Amazon, Apple, Baidu und Google stammen. Diese Investitionen werden im Bericht auf 18 bis 27 Milliarden US Dollar geschätzt.

Der Bericht des McKinsey Global Institute (2017) zeigt weiter, dass viele der leitenden Angestellten, der über 3'000 anderen weltweit befragten Unternehmen, aktuell nicht wissen, welche Vorteile die künstliche Intelligenz für ihr Unternehmen bieten könnte. Aus diesem Grund bleiben Investitionen von Firmen ausserhalb des Technologie-Sektors aus.

Im Folgenden werden drei Anwendungsbeispiele der künstlichen Intelligenz aufgezeigt. Diese Beispiele sollen verdeutlichen, dass auch bei Unternehmen ausserhalb des Technologie-Sektors eine Investition in die künstliche Intelligenz sinnvoll ist.

---

<sup>3</sup>Dieser Test wurde auf dem ImageNet Datensatz zur Objekterkennung durchgeführt. Mehr zur Objekterkennung auf Bildern im Kapitel 4.4.1.

#### **2.4.1 Ping An Insurance Co. of China Ltd.**

Ping An Insurance Co. of China Ltd. hatte bereits im Jahr 2017 110 Data Scientists eingestellt, welche unter anderem im Bereich der künstlichen Intelligenz tätig sind. Diese Data Scientists haben mehr als 30 CEO gesponserte Projekte in diesem Bereich durchgeführt (Ransbotham et al., 2017). Der technologische Fortschritt des Ping An Konzerns zeigt sich auch auf der dedizierten Webseite von Ping An Technology, der Konzern eigenen IT Unternehmung. Die Webseite stellt bereits auf der Startseite die künstliche Intelligenz als eine der wichtigsten Technologien für die Unternehmung vor. Es werden auch einige Anwendungsfälle aufgezeigt. So gibt Ping An Technology an, Systeme entwickelt zu haben, welche eine Grippe, Diabetes und andere Krankheiten vorhersagen können (Ping An Technology, o.D.).

#### **2.4.2 Blue River Technology**

Blue River Technology, heute Teil von John Deer, hat ein System entwickelt, welches mit Hilfe von Computer Vision und künstlicher Intelligenz beurteilen kann, ob eine Pflanze von einem Schädling befallen ist. Das System kann berechnen, wie viel von welchem Pestizid notwendig ist, um den Schädling zu bekämpfen. Betrachtet man die Nebenwirkungen solcher Pestizide und deren aktuell massenhaften Einsatz, so ist dieses System ein wichtiger Bestandteil der Nahrungssicherung (Marr, o.D. b).

#### **2.4.3 Infervision**

Infervision, ein chinesisches Start-up, nutzt Computer Vision und künstliche Intelligenz, um die Lebensqualität zu verbessern. Aufgrund mangelnder Radiologen, um Röntgenbilder auf Lungenkrebs zu prüfen, setzte sich das Start-up zum Ziel, die Röntgenbilder von einem Computer beurteilen zu lassen. Das entwickelte System ermöglicht es den wenigen Radiologen in China, Röntgenbilder genauer und effizienter zu verarbeiten. Dies ist für die Heilung der Krankheit, welche jährlich mehr als 600'000 Chinesen das Leben kostet, sehr wichtig (Marr, o.D. a).



## 3 | Künstliche Intelligenz: Grundlagen und Ansätze in der Praxis

Die künstliche Intelligenz ist ein in der Literatur oft diskutiertes Themengebiet. Bereits 2009 geben Russell und Norvig (2009) auf über 1000 Seiten einen noch immer aktuellen und sehr umfangreichen Überblick. Weiter vertiefen die beiden Autoren viele Teilgebiete der künstlichen Intelligenz und erläutern grundlegende Konzepte ausführlich.

Es kann gesagt werden, dass in der Grundlagenforschung zur künstlichen Intelligenz bereits viele Forschungsergebnisse vorliegen. Es werden etliche, etablierte und experimentelle, Techniken diskutiert und täglich weiterentwickelt. Zur Automatisierung eines Geschäftsprozesses und somit für die Entwicklung eines Prototypen für die AXA Gesundheitsvorsorge stehen unzählige Möglichkeiten zur Verfügung.

Dieses Kapitel gibt ein Überblick über das Themengebiet der künstlichen Intelligenz. Es werden für diese Arbeit relevante Techniken, Konzepte und Begriffe erläutert. Es werden Konzepte und Metriken vorgestellt, welche verwendet werden, um eine künstliche Intelligenz zu bewerten. Es wird aufgezeigt, wie eine künstliche Intelligenz modelliert wird und die finale Architektur entsteht. Zum Schluss wird die end-to-end Entwicklung und der Betrieb einer künstlichen Intelligenz diskutiert.

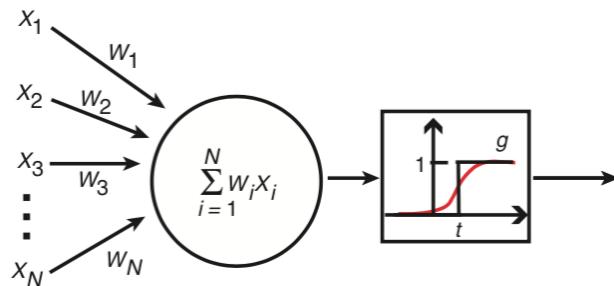
### 3.1 Neuronale Netzwerke

Abgesehen von Rechenaufgaben sind Menschen leistungsfähiger als Computer. Wir sind beispielsweise in der Lage, Gesichter zu erkennen oder in einem dunklen Raum Personen anhand Ihrer Stimme zu identifizieren. Der interessanteste Unterschied des menschlichen Gehirns zu einem Computer ist der Fakt, dass unser Gehirn lernt, ohne eine Softwareaktualisierung zu erhalten. Wir brauchen nicht erst eine neue Software, um zu lernen, ein Fahrrad zu fahren (Krogh, 2008). Doch wie funktioniert das?

Die Berechnungen des menschlichen Gehirn werden durch hoch vernetzte Neuronen gemacht. Dabei interagieren die Neuronen mit Stromimpulsen durch die neuronale Verkabelung, bestehend aus Nervensäulen, Synapsen und Zellfortsätzen. 1943 modellierten McCulloch und Pitts Neuronen als Schalter, welche aufgrund der eingehenden Signale ein- oder ausgeschaltet werden. Die Gewichtung der eingehenden Signale sind dabei die Synapsen. Aus diesem Modell entstand das Konzept von neuronalen Netzwerken (Krogh, 2008).

Ein Neuron wurde von McCulloch und Pitts als eine Threshold Unit (vgl. Abbildung 2) modelliert, welche Eingabewerte anderer Units oder externer Quellen erhält. Die Threshold Unit erhält  $N$  Eingangssignale  $x_1, \dots, x_N$ . Diese Eingangssignale werden mit dem zugehörigen Gewicht  $w_1, \dots, w_N$  multipliziert und schlussendlich summiert. Je nach Modell wird die Aktivierung, sprich der Ausgabewert, des Neurons durch das Summenprodukt auf zwei verschiedene Arten berechnet. Zum einen kann das Neuron je nach Erreichung eines gewissen Thresholds ( $t$ ) mit 0 oder 1 aktiviert werden. Andererseits kann eine Aktivierungsfunktion verwendet werden. Das Modell von McCulloch und Pitts in Abbildung 2 zeigt einen kontinuierlichen Sigmoiden  $\sigma(x) = \frac{1}{1+e^{-x}}$  (rote Linie) als mögliche Aktivierungsfunktion (Krogh, 2008).

Abbildung 2: Modell eines Neurons nach McCulloch und Pitts



Quelle: Krogh (2008)

Neuere Modelle, welche heute in neuronalen Netzwerken zur Anwendung kommen, verwenden stets eine Aktivierungsfunktion anstelle eines Thresholds. Der Begriff Threshold Unit ist aus diesem Grund nicht mehr geläufig. Die Rectified Linear Unit Aktivierungsfunktion (kurz ReLU) ist aktuell die verbreitetste. Die ReLU Aktivierungsfunktion bedient sich selbst auch einem Threshold. Sie begrenzt das Summenprodukt aus den Eingangssignalen und Gewichten gegen unten auf 0 und wird als  $f(x) = \max(0, x)$  definiert. Im Gegensatz zur Aktivierung aufgrund eines Thresholds kann der Ausgabewert somit einen kontinuierlichen Wert von 0 bis  $\infty$  annehmen. Im Gegensatz zu einem kontinuierlichen Sigmoiden hat die ReLU Aktivierungsfunktion nicht nur den Vorteil, dass sie weniger rechenintensiv ist, sondern dass das neuronale Netzwerk bis zu sechs mal effektiver trainiert werden kann. Der Nachteil dabei ist, dass durch ein zu schnelles Training Neuronen mit der ReLU Aktivierungsfunktion Gewichte erlernen können, durch welche immer ein Ausgabewert von 0 resultiert. Solche Neuronen werden als tote Neuronen bezeichnet. Durch eine kleinere Learning Rate<sup>4</sup> kann dieses Phänomen vermindert werden (Karpathy, 2015b).

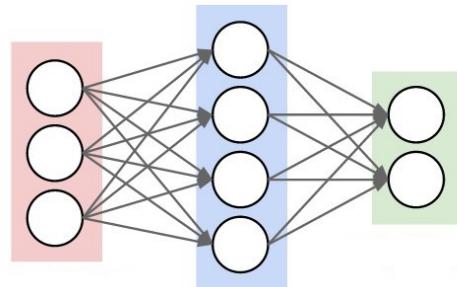
Neben den trainierbaren Gewichten wurden neuere Modelle von Neuronen auch um einen trainierbaren Bias erweitert. Dieser Bias wird zum Summenprodukt der Eingabesignale und der Gewichte addiert, bevor dieses der Aktivierungsfunktion übergeben wird. Ist die Aktivierungsfunktion  $f$  gegeben, so wird der Ausgabewert eines Neurons als  $f(\sum_i^n w_i x_i + b)$  definiert (Karpathy, 2015b).

Neuronale Netzwerke werden als eine Ansammlung von vernetzten Neuronen modelliert. Die Ausgabe eines Neurons fliesst als Eingabe in das nächste Neuron. Wichtig dabei ist, dass keine Schlaufen erlaubt sind, da dies in einer unendlichen Schlaufe resultieren würde. Die Neuronen sind nicht willkürlich angeordnet, sondern in Schichten organisiert. Für normale neuronale Netzwerke ist ein sogenannter Fully Connected Layer (auch Dense Layer genannt) die verbreitetste Art von Schicht. Jedes Neuron einer solchen Schicht ist mit jedem Neuron aus der vorherigen Schicht verbunden. Die Neuronen innerhalb einer Schicht haben keine Verbindungen untereinander. Die Abbildung 3 zeigt ein zweischichtiges neuronales Netzwerk mit drei Eingabewerten (rot), einer versteckten Schicht (englisch Hidden Layer) mit vier Neuronen (blau) und einer Ausgabe-Schicht mit zwei Neuronen (grün) (Karpathy, 2015b).

---

<sup>4</sup>Die Learning Rate definiert, wie stark die trainierbaren Parameter (beispielsweise die Gewichte) eines Neurons nach jedem Trainingsdurchgang angepasst werden. Die Learning Rate bestimmt somit die Geschwindigkeit, mit welcher ein neuronales Netzwerk lernt (Goodfellow, Bengio & Courville, 2016).

Abbildung 3: Modell eines neuronalen Netzwerks mit zwei Fully Connected Layer



Quelle: Karpathy (2015b)

Damit ein neuronales Netzwerk leistungsfähig werden kann, muss es ähnlich wie ein Mensch, erst lernen. Für das neuronale Netzwerk bedeutet Lernen, für die Gewichte und Bias geeignete Werte zu finden. Dieses computersimulierte Lernen, auch Machine Learning genannt, funktioniert dabei so, dass für die Gewichte der Verbindungen, sprich die Stärke der Synapsen, ein zufälliger Wert gewählt wird. Anschliessend wird vom Netzwerk eine Übungsaufgabe gelöst. Das Resultat des Netzwerks ist zu Beginn höchstwahrscheinlich falsch und die Gewichte und Bias werden mit einem kleinen Schritt angepasst. Es werden dann immer weitere Aufgaben gelöst und die Gewichte entsprechend angepasst bis das Netzwerk die gewünschten Resultate liefert. Dieses Training kann durch verschiedene Algorithmen implementiert werden. Einer solcher Algorithmus wird im Kapitel 3.3 beschrieben (Krogh, 2008).

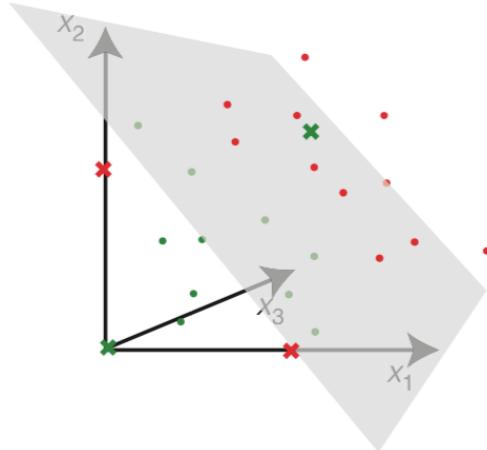
Seit mehreren Jahren liefern neuronale Netzwerke bessere Resultate als klassische Techniken. Dabei werden neuronale Netzwerk vorwiegend für visuelle Aufgaben und immer häufiger auch zur Verarbeitung natürlicher Sprache verwendet (Olah, 2014).

## 3.2 Tiefe neuronale Netzwerke

Neuronale Netzwerke finden oftmals bei Klassifizierungsproblemen Anwendung. Dabei soll aufgrund bestimmter Eingabewerte eine Klasse bestimmt werden. Ein Beispiel dafür ist die Klassifizierung eines Säugetiers in die Klasse Hund oder Katze aufgrund ihrer Merkmale (Krogh, 2008).

Einfache Netzwerke von Neuronen können Klassifizierungsprobleme dann Lösen, wenn die Klassen linear separierbar sind. Die Abbildung 4 veranschaulicht die lineare Separierung mit Hilfe einer Ebene in einem dreidimensionalen Raum. Die Ebene separiert die grünen und roten Punkte voneinander (Krogh, 2008).

Abbildung 4: Konzept der linearen Separierbarkeit

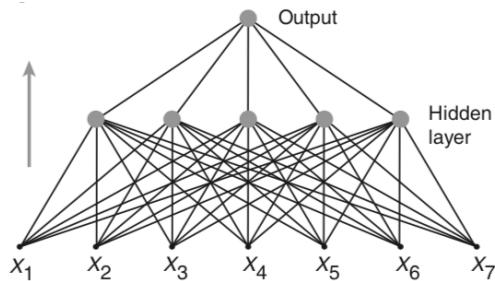


Quelle: Krogh (2008)

Der Abbildung 4 ist zu entnehmen, dass die Problemstellung im abgebildeten Fall, wie die meisten Klassifizierungsprobleme, nicht linear separierbar ist. Die roten und grünen Kreuze markieren dabei Punkte, welche auf der falschen Seite der Ebene liegen. Eine Klassifizierung durch ein einfaches Netzwerk von Neuronen würde die Klasse dieser Datensätze falsch vorhersagen (Krogh, 2008).

Um das Modell zur Klassifizierung zu verbessern, können zusätzliche Ebenen in den dreidimensionalen Raum eingesetzt werden. Durch eine weitere Ebene ist es dem Modell möglich, mehr Datensätze korrekt zu klassifizieren. Neue Ebenen werden mit neuen Schichten von Neuronen modelliert (vgl. Abbildung 5). Diese neuen Schichten, welche sich zwischen den Eingabe und Ausgabe Schichten befinden, werden versteckte Schichten (Hidden Layers) genannt. Ein Modell mit mindestens einer solcher versteckter Schicht wird als tiefes neuronales Netzwerk (Deep Neural Network) bezeichnet (Krogh, 2008).

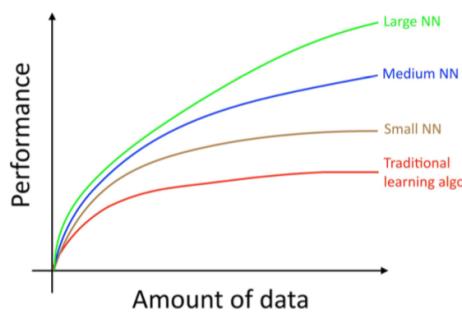
Abbildung 5: Modell eines tiefen neuronalen Netzwerks mit einer versteckten Schicht



Quelle: Krogh (2008)

Mit immer mehr verfügbarer Rechenkapazität und immer mehr Daten steigt die Popularität tiefer neuronaler Netzwerke gegenüber traditionellen Lernalgorithmen wie der logistischen Regression. Je tiefer und breiter ein neuronales Netzwerk ist und je mehr Trainingsdaten zur Verfügung stehen, desto besser wird es eine Problemstellung lösen können (vgl. Abbildung 6) (Ng, 2018).

Abbildung 6: Verhältnis der verfügbaren Daten und der Genauigkeit unterschiedlich grosser neuronaler Netze



Quelle: Ng (2018)

Ein tiefes neuronales Netzwerk stellt neue Anforderungen an das Training. Ein Netzwerk mit versteckten Schichten kann nicht mehr auf eine analytische Weise trainiert werden. Aus diesem Grund wird ein komplexerer Lernalgorithmus benötigt. Ein solcher Algorithmus wird im Kapitel 3.3 erläutert (Krogh, 2008).

Auch ist ein tiefes neuronales Netzwerk anfälliger auswendig zu lernen und birgt somit die Gefahr schlecht zu generalisieren. Dieses sogenannte Overfittig wird im Kapitel 3.4 erläutert.

### 3.3 Backpropagation

Da tiefe neuronale Netzwerke zu komplex sind, um mit einem analytischen Ansatz trainiert zu werden, muss ein anderes Vorgehen gefunden werden. Das am meisten angewendete Vorgehen zum Training von tiefen neuronalen Netzwerken ist die sogenannte Backpropagation. Bei der Backpropagation werden zu Beginn zufällige Gewichte und Bias festgesetzt. Das Netzwerk löst mit diesen zufälligen Gewichten und Bias eine erste Übungsaufgabe. Die Abweichung vom erwarteten Resultat (der sogenannte Fehler) wird anschliessend quadriert. Ziel des Backpropagation ist es, diesen quadrierten Fehler zu minimieren. Dies wird erzielt, indem das Verfahren des steilsten Abstiegs, auch Gradientverfahren (englisch gradient descent), angewendet wird. Mit Hilfe dieses Verfahrens, zur Lösung allgemeiner Optimierungsprobleme aus der Numerik, werden nun die Gewichte und das Bias so angepasst, dass der quadrierte Fehler minimiert werden kann. Dieses Vorgehen wird mit weiteren Übungsaufgaben wiederholt, bis sich der quadrierte Fehler nicht mehr verändert (Krogh, 2008).

Bei der Backpropagation müssen einige Herausforderungen beachtet werden. So kann durch das Gradientverfahren nur ein lokales Minimum gefunden werden. Ob dieses lokale Minimum dem globalen Minimum entspricht, ist nicht bekannt. Das Ergebnis des Trainings ist damit von den zufällig gewählten Startwerten der Gewichte und Bias abhängig (Krogh, 2008).

Die grösste Problematik beim Trainieren von neuronalen Netzwerken, besonders von tiefen neuronalen Netzwerken, ist die Gefahr des sogenannten Overfitting oder Auswendiglernens (Krogh, 2008).

### 3.4 Over- und Underfitting

Eine Herausforderung der künstlichen Intelligenz beziehungsweise des Machine Learning ist es, auf zuvor unbekannten Daten gute Ergebnisse zu erzielen. Diese Fähigkeit wird als Generalisierung bezeichnet (Goodfellow et al., 2016).

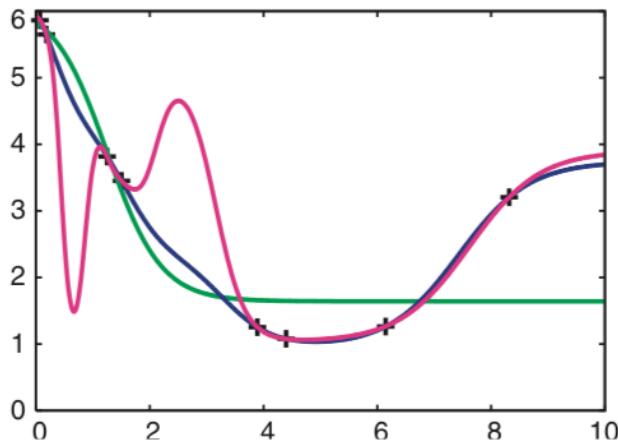
Um eine Generalisierung sicherzustellen, wird ein Modell immer auf einem Trainingsdatensatz trainiert und auf einem Testdatensatz validiert. Ziel eines Modells ist es, die Fehler während dem Training (auch Bias genannt, nicht aber mit dem trainierbaren Bias eines Neurons zu verwechseln) zu minimieren und den Unterschied zwischen der Fehlerquote während dem Training und während dem Testen (auch Varianz genannt) möglichst klein zu halten. Die Nichterreichung dieser Ziele wird Over- respektive Underfitting genannt (Goodfellow et al., 2016).

Die Problematik des Underfitting wurde bereits bei der Einführung von tiefen neuronalen Netzwerken, im Kapitel 3.2, angeschnitten. Ein Modell, welches zu wenige Parameter hat, ist nicht in der Lage eine komplexe Problemstellung abzubilden. Das Modell hat somit einen hohen Bias (hohe Fehlerquote auf dem Trainingsdatensatz). In diesem Fall ist es wichtig, das Modell anzupassen. Eine Erhöhung der Anzahl Trainingsdatensätze hilft nicht, den Bias zu reduzieren (Ng, 2018).

Das sogenannte Overfitting, oder auch Auswendiglernen, bezeichnet die Problematik, dass sich ein Modell aufgrund zu vieler Parameter bei zu wenig Trainingsdaten zu stark an diese Trainingsdaten anpasst. Das Modell kann die Trainingsdaten mit annähernd 100% Trefferquote beurteilen, hat also ein kleines Bias. Soll das Modell dann aber einen neuen Datensatz beurteilen, so sinkt die Trefferquote erheblich. Das Modell hat eine hohe Varianz und ist nicht in der Lage das gelernte zu generalisieren (Ng, 2018; Krogh, 2008).

Die Problematik des Overfitting ist aus der Mathematik bekannt. Hat eine Funktion zu viele freie Parameter, so passt sie sich zu stark an die vorgegebenen Punkte an. Die Abbildung 7 veranschaulicht diese Problematik anhand von Graphen von Funktionen, welche 8 Punkte fitten sollen. Der grüne Graph ist ein Beispiel des Underfitting. Er hat zu wenige Parameter, um eine tiefe Abweichung zu erreichen. Der pinke Graph zeigt ein Beispiel eines Overfitting. Mit vielen Parametern vermag der Graph alle Punkte perfekt zu schneiden. Die vielen extremen Wendepunkte deuten aber darauf hin, dass der Graph neue Punkte mit hoher Wahrscheinlichkeit nicht schneiden würde. Der blaue Graph gilt als Beispiel für ein gutes Fitting. Der Graph ist nahe an den Punkten, ohne dabei extreme Wendepunkte haben zu müssen (Krogh, 2008).

Abbildung 7: Over- und Underfitting dargestellt anhand von Graphen von Funktionen



Quelle: Krogh (2008)

Während die Problematik des Overfitting aus anderen Bereichen bereits bekannt ist, scheinen neuronale Netzwerke besonders anfällig für eine solche Überparametrierung. Würden wir ein Modell entwickeln, welches anhand von 20 Merkmalen (20 Eingabewerte) mit Hilfe einer versteckten Schicht von 10 Neuronen erkennen soll, ob es sich um einen Hund oder eine Katze handelt, so würden 221 Parameter geschaffen. Jeder der 20 Eingabewerte wird durch ein Gewicht mit den 10 Neuronen aus der versteckten Schicht verbunden ( $10 * 20 = 200$  Parameter). Jedes dieser Neuronen hat einen Bias (1 Parameter) und ist mit dem Ausgabeneuron verknüpft (1 Parameter). Das Ausgabeneuron selbst hat auch wieder einen Bias (1 Parameter). Wird dieses Modell mit 221 Parametern nun mit Hilfe von nur 100 Trainingsdatensätzen trainiert, so kann es diese problemlos auswendig lernen. Das Modell kann auf neuen Datensätzen nicht oder nur schlecht generalisieren und wird somit eine hohe Varianz aufweisen (Krogh, 2008).

Um ein solches Overfitting zu verhindern stehen diverse Techniken zur Verfügung. Eine beliebte Regularisierungstechnik ist es, eine sogenannte Dropout Schicht in das Netzwerk einzubringen. Diese Dropout Schicht deaktiviert während dem Training zufällige Neuronen. Dies hat zur Folge, dass während dem Training nur Teile des Netzwerks trainiert werden<sup>5</sup> (Goodfellow et al., 2016).

Um Over- respektive Underfitting zu erkennen, ist es wichtig, ein neuronales Netzwerk an Daten zu testen, welche nicht zum Training verwendet wurden (Krogh, 2008).

## 3.5 Convolutional Neural Network

Convolutional Neural Network (kurz CNN) funktionieren ähnlich wie normale neuronale Netzwerke. Sie bestehen ebenfalls aus Neuronen mit trainierbaren Gewichten und Bias. Convolutional Neural Networks machen eine Annahme zu ihren Eingabewerten. Sie sind speziell auf Bilder ausgelegt. Durch diese Annahme können gewisse Verhalten direkt in die Architektur einprogrammiert werden. Dadurch kann die Anzahl trainierbarer Parameter gegenüber normalen neuronalen Netzwerken reduziert und die Performance erhöht werden (Karpathy, 2015a).

Die Eingabe in ein Convolutional Neural Network ist ein dreidimensionaler Vektor im Format Format Breite x Höhe x 3. Dieser Vektor hält die rohen Pixeldaten des Bildes. Ein Bild in einer Auflösung von 32x32 Pixeln wird durch einen Vektor der Grösse 32x32x3 repräsentiert. Die dritte Dimension, auch Tiefe genannt, hält die Farbinformationen eines Pixels im RGB Format (Karpathy, 2015a).

---

<sup>5</sup>Die genaue Funktionsweise einer Dropout Schicht ist für diese Arbeit nicht relevant, kann aber bei Bedarf in Goodfellow et al. (2016) nachgelesen werden.

Convolutional Neural Networks bestehen hauptsächlich aus den folgenden drei verschiedenen Typen von Schichten (Karpathy, 2015a):

- **Convolutional Layer:** Berechnen Ausgabewerte aufgrund von Verbindungen zu Ausschnitten aus der vorhergehenden Schicht, sprich Ausschnitten aus dem Bild, welches als Eingabewert verwendet wird.
- **Pooling Layer:** Reduziert den mehrdimensionalen Vektor aus der vorhergehenden Schicht entlang der räumlichen Dimensionen (Breite und Höhe).
- **Fully Connected Layer:** Ein Fully Connected Layer, wie er aus dem normalen neuronalen Netzwerk bekannt ist.

### Convolution Layer

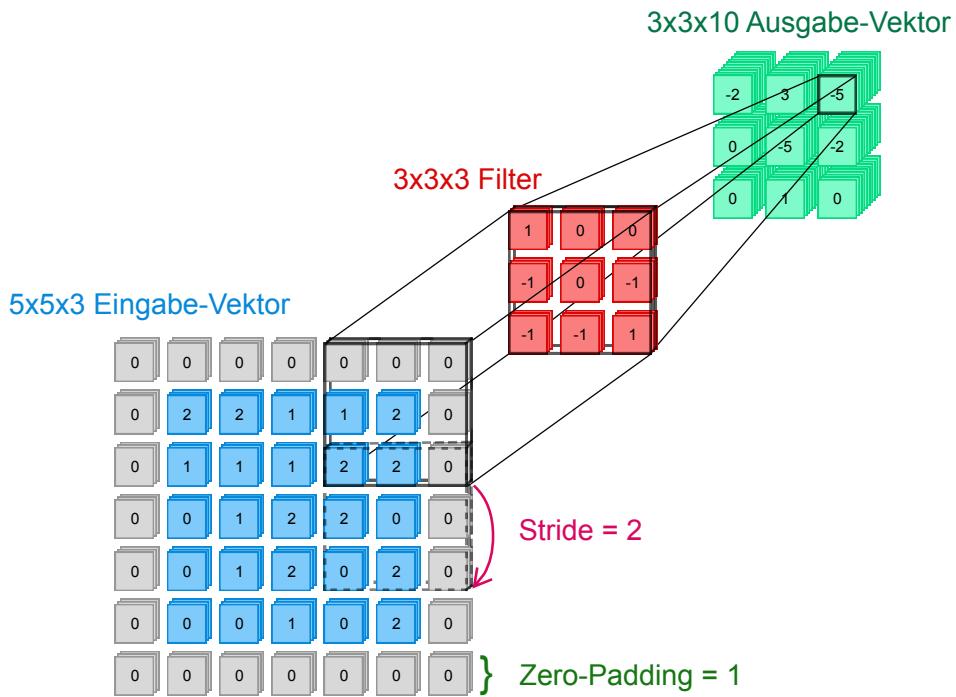
Ein Convolution Layer besteht aus mehreren, trainierbaren Filtern. Diese Filter beziehen sich auf einen räumlich (Breite und Höhe) kleinen Teil des Eingabe-Bildes respektive des Eingabe-Vektors. Die Filter haben die gleiche Tiefe (Größe der dritten Dimension) wie der Eingabe-Vektor. Ein Filter eines Convolutional Neural Network könnte beispielsweise die Größe 5x5x3 aufweisen. In diesem Fall ist der Filter mit einem 5x5 Pixel grossen Ausschnitt des Eingabe-Vektors verknüpft. Dieser Filter wandert horizontal und vertikal über den Eingabe-Vektor. Dabei wird jeweils das Skalarprodukt der Werte des Filters und der Werte des Eingabe-Vektors berechnet. Die Skalarprodukte, welche so an verschiedenen Positionen des Eingabe-Vektors berechnet werden, bilden einen zweidimensionalen Aktivierungs-Vektor. Während dem Training erlernt das Netzwerk so Filter, welche durch bestimmte visuelle Merkmale (Kanten, Ecken, Häufungen von Farben) aktiviert werden. Jeder Convolution Layer hat mehrere solche Filter, damit das Netzwerk auf mehrere visuelle Merkmale reagieren kann. Pro Filter entsteht ein zwei-dimensionaler Aktivierungs-Vektor, welche gestapelt werden und dadurch den drei-dimensionalen Ausgabe-Vektor ergeben. Die Abbildung 8 veranschaulicht dieses Konzept. Auf der linken Seite ist der Eingabe-Vektor in blau dargestellt. Auf diesem Vektor wird der in rot dargestellte Filter angewendet. Dieser Filter deckt einen kleinen räumlichen Teil aber die gesamte Tiefe des Bildes ab. Die Aktivierung des Filters wird im grünen Ausgabe-Vektor abgelegt (Karpathy, 2015a).

Die Anzahl der Filter und die Art wie sich ein Filter über die räumlichen Dimensionen des Eingabe-Vektors bewegt, kann durch Hyperparameter<sup>6</sup> konfiguriert werden. Mit der Anzahl an Filtern kann konfiguriert werden, wie viele verschiedene visuelle Merkmale der Convolution Layer erkennen kann. Die Anzahl an Filtern bestimmt die Tiefe des Ausgabe-Vektors. Die Abbildung 8 zeigt einen Convolution Layer mit zehn Filtern. Die dritte Dimension des Ausgabe-Vektor hat dadurch die Größe zehn (Karpathy, 2015a).

---

<sup>6</sup>Als Hyperparameter werden jene Parameter eines Machine Learning Modells bezeichnet, welche nicht während dem Training gelernt, sondern von aussen in das Modell hineingegeben werden. Die Hyperparameter können als Konfiguration eines Modells verstanden werden (Román Aragay, 2018).

Abbildung 8: Visualisierung eines Convolution Layer



Quelle: In Anlehnung an Karpathy (2015a)

Die Art wie sich ein Filter über die räumlichen Dimensionen bewegt, kann durch das sogenannte Zero-Padding und den Stride konfiguriert werden. Das Zero-Padding sagt aus, wie viele zusätzliche Neuronen um den Eingabe-Vektor herum gelegt werden sollen. Das Zero-Padding ermöglicht, dass die Filter auch visuelle Merkmale an den Rändern des Eingabe-Vektors erkennen können. Der Stride bestimmt die Anzahl Neuronen, um welche der Filter nach jeder Anwendung verschoben wird. In Abbildung 8 ist ein Beispiel eines Convolution Layer mit einem Zero-Padding von eins und einem Stride von zwei zu sehen. Durch diese Hyperparameter und die Grösse der ersten beiden Dimensionen des Eingabe-Vektors wird die Grösse der ersten beiden Dimensionen des Ausgabe-Vektors bestimmt. Bei der Wahl des Zero-Padding und des Strides muss darauf geachtet werden, dass die Filter genau auf den Eingabe-Vektor passen. Würde im Beispiel, welches in Abbildung 8 ersichtlich ist, ein Stride von drei anstelle zweи gewählt werden, so könnte der Filter nur 1.33 mal verschoben werden und der Convolution Layer wäre nicht funktionsfähig (Karpathy, 2015a).

Wird in einer solchen Schicht jeder Filter auf jeder Position einzeln trainiert, so entstehen enorm viele trainierbare Parameter. Dies zeigt ein Beispiel eines Bildes der Grösse 227x227 Pixel und 96 Filter der Grösse 11x11 bei einem Stride von vier und Zero-Padding von null. Jeder Filter kann auf  $1 + (227 - 11)/4 = 55$  horizontal und 55 vertikal unterschiedlichen Positionen auf dem Bild angewendet werden. In diesem Fall entsteht ein Ausgabe-Vektor der Grösse 55x55x96. Die  $55 * 55 * 96 = 290'400$  Neuronen dieses Ausgabe-Vektors werden nun jeweils durch  $11 * 11 * 3 = 363$  Gewichte und einen Bias berechnet. Dadurch entstehen  $290'400 * 363 = 105'705'600$  trainierbare Parameter. Dieses Beispiel zeigt, dass dadurch bereits bei kleinen Bildern trotz hohem Stride eine grosse Anzahl an trainierbaren Parametern entsteht (Karpathy, 2015a).

Durch sogenanntes Parameter Sharing kann die Anzahl der trainierbaren Parameter stark reduziert werden. Das Parameter Sharing basiert auf der Annahme, dass wenn ein Filter an einer räumlichen Position (x,y) angewendet werden kann, um ein visuelles Merkmal zu erkennen, dies auch für eine andere Position (x2,y2) gilt. Das bedeutet, dass ein Filter an jeder Position des Eingabe-Vektors die selben Gewichte und Bias verwendet. Durch diese Annahme können die Parameter im oben genannten Beispiel auf  $11 * 11 * 3 = 363$  Gewichte und ein Bias pro Filter reduziert werden. Dadurch wird die gesamte Anzahl an trainierbaren Parametern in diesem Beispiel auf  $96 * (363 + 1) = 34'944$  reduziert. Durch diese Annahme eignen sich Convolution Neural Networks besonders gut für Problemstellungen mit Bildern (Karpathy, 2015a).

#### **Pooling Layer**

Nach einem oder teilweise mehreren Convolution Layern ist es üblich, einen Pooling Layer anzubringen. Dieser Pooling Layer reduziert die räumliche Dimension des Eingabe-Vektors. Eine der häufigsten Anwendungen ist ein Max Pooling Layer der Grösse 2x2 mit einem Stride von zwei. Dieser Max Pooling Layer reduziert jeden betrachteten 2x2 Ausschnitt aus dem Eingabe-Vektor auf das grösste Neuron. Dadurch halbiert er den Eingabe-Vektor horizontal sowie vertikal und reduziert somit die Anzahl Neuronen um 75%. Anstelle von Max Pooling Layern kommen auch andere Verfahren wie der Average Pooling Layer zur Anwendung. Der Max Pooling Layer liefert meist bessere Resultate als ein Average Pooling Layer (Karpathy, 2015a).

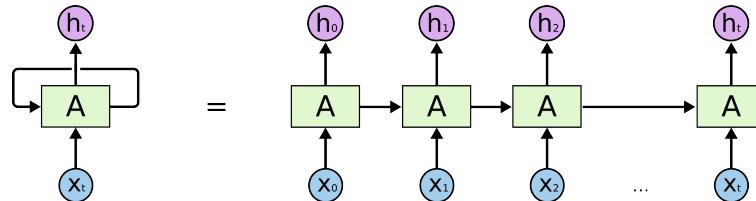
Pooling Layer kommen zur Anwendung, damit die Anzahl der trainierbaren Parametern und somit der Rechenaufwand tief gehalten werden kann. Pooling Layer helfen auch Overfitting zu vermeiden (Karpathy, 2015a).

## 3.6 Long-Short-Term-Memory Netzwerke

Menschen starten ihren Denkprozess nicht jede Sekunde von Neuem. Beim Lesen wird jedes Wort aufgrund des Verständnisses des vorherigen Wortes verstanden. Gedanken sind persistent. Genau solche persistenten Gedanken sind mit den bisherigen Ansätzen für neuronale Netze nicht modellierbar. Hier kommen Recurrent Neural Networks (kurz RNN) ins Spiel. Recurrent Neural Networks sind neuronale Netzwerke mit integrierten Schlaufen (Olah, 2015).

In Abbildung 9 wird ein RNN mit einer Schlaufe dargestellt. Daneben ist dasselbe Netzwerk in einer anderen, ausgerollten Weise zu sehen. Die Abbildung verdeutlicht, wie Informationen innerhalb einer Schicht von einem Neuron zum anderen fliessen können. Mit diesem Informationsfluss von Neuron zu Neuron werden die Resultate jeweils von den vorhergehenden Neuronen beeinflusst. Damit wird eine Art Kurzzeitgedächtnis geschaffen, welches dem neuronalen Netzwerk erlaubt mit Kontextinformationen zu arbeiten (Olah, 2015).

Abbildung 9: Informationsfluss durch ein Recurrent Neural Network

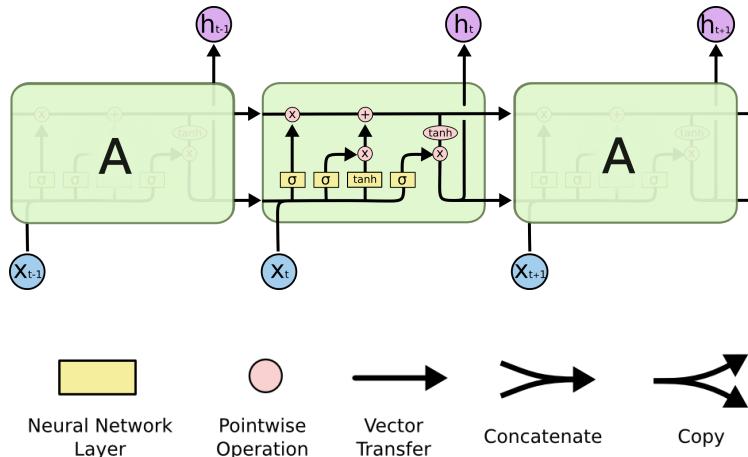


Quelle: Olah (2015)

Ein Problem von Recurrent Neural Networks ist, dass nur ein Kurzzeitgedächtnis zur Verfügung steht. Liegen Informationen länger zurück, sprich der Abstand zwischen den beiden relevanten Neuronen ist zu gross, gehen diese Informationen verloren. Eine Lösung für diese Problematik bieten Long-Short-Term-Memory (kurz LSTM) Netzwerke. Diese Spezialform von Recurrent Neural Networks arbeitet mit sogenannten Gates, um zu regulieren, wie viel Kontextinformationen behalten oder vergessen werden sollen. Mit vier solchen Gates, bestehend aus einem Neural Network Layer und einer Pointwise Operation (vgl. Abbildung 10), ist ein LSTM Netzwerk in der Lage, nicht nur ein Kurz- sondern auch ein Langzeitgedächtnis aufzubauen (Olah, 2015).

Recurrent Neural Networks wurden in der Vergangenheit sehr erfolgreich für viele Aufgaben, wie beispielsweise die Spracherkennung, Sprachmodellierung, maschinelle Übersetzung sowie Objektkennung, angewendet. In den meisten Fällen wurden dabei LSTM Netzwerke angewendet, da die Resultate um ein Vielfaches besser ausfallen als mit herkömmlichen RNNs (Olah, 2015).

Abbildung 10: Informationsfluss eines LSTM Netzwerk



Quelle: Olah (2015)

LSTM Netzwerke sind für diese Arbeit besonders zur Texterkennung und Rechtschreibkorrektur interessant.

### 3.7 Texterkennung

Ein wichtiger Bestandteil des Prototypen zur Indexierung von Rechnungen ist die Erkennung von Texten, in Druckbuchstaben oder Handschrift, auf Rechnungen. Die erkannten Texte bilden die Grundlage für jegliche digitale Verarbeitung der Rechnungen.

Die herkömmliche Feature-detection in Texterkennungssoftware wird immer mehr mit künstlicher Intelligenz ersetzt. Neuberg (2017) beschreibt wie Dropbox künstliche Intelligenz anwendet, um Texte aus Fotografien von Dokumenten durchsuchbar zu machen. Zur Anwendung kommen dabei verschiedene Techniken aus dem Bereich der künstlichen Intelligenz: Convolutional Neural Networks, Long-Short-Term-Memory Netzwerke, Connectionist Temporal Classification<sup>7</sup> (CTC) und weitere (Neuberg, 2017).

<sup>7</sup>Connectionist Temporal Classification ist ein Konzept aus dem Training von neuronalen Netzwerken, welches vor allem in der Handschrifterkennung Verwendung findet. Dabei wird eine spezielle Trainingsfunktion verwendet, durch welche die Positionierung von Buchstaben generalisiert und somit der Lernprozess des neuronalen Netzwerks vereinfacht werden kann (Scheidl, 2018).

### 3.8 Word embedding

---

Auch die Texterkennungssoftware Tesseract, welche ursprünglich als Forschungsprojekt im HP Lab entwickelt wurde und seit 2005 als Open Source Software zur freien Verfügung steht, verwendet seit Version 4 künstliche Intelligenz (Smith, 2007). So wurde die Feature-detection durch ein LSTM Netzwerk mit mehr als 100 Schichten ersetzt. Die Texterkennung konnte so nicht nur qualitativ stark verbessert werden, sondern ist auch schneller als zuvor. Doch auch nach den Verbesserungen müssen die Ergebnisse fallspezifisch optimiert werden (O.V., 2018a, 2018b).

## 3.8 Word embedding

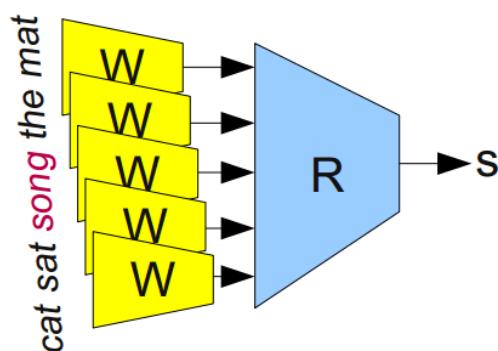
Neuronale Netzwerke funktionieren mit Zahlen. Damit auch Wörter und ganze Sätze von solchen Netzwerken verarbeitet werden können, muss eine geeignete Repräsentation von Wörtern durch Zahlen gefunden werden. Der Prozess, mit welchem ein Wort in eine zahlenbasierte Repräsentation gebracht wird, nennt sich Word embedding (Olah, 2014).

Die ersten Word embeddings wurden von Bengio, Ducharme und Vincent (2001) eingeführt. Trotz des schon fortgeschrittenen Alters dieses Forschungsgebietes ist es noch immer von neuen Entwicklungen geprägt (Olah, 2014).

Technisch gesehen ist ein Word embedding eine parametrierte Funktion, welche Wörter einer bestimmten Sprache in einen hochdimensionalen Vektor (typischerweise 200-500 Dimensionen) transformiert. Um diese hoch komplexe Funktion zu definieren, kommt Machine Learning zum Einsatz. Olah (2014) beschreibt in seinem Artikel ein Beispiel, bei welchem eine solche Word embedding Funktion trainiert wird. Die Resultate aus dem Word embedding werden in ein neuronales Netzwerk zur Prüfung eines 5-Grammes gespiesen und dann das Gesamtkonstrukt trainiert. Die Abbildung 11 veranschaulicht dieses Vorgehen mit fünf Ausprägungen der Word embedding Funktion  $W$  und dem neuronalen Netzwerk  $R$  (Olah, 2014).

Um sich Word embeddings besser vorstellen zu können, werden im Folgenden zwei verschiedene Möglichkeiten zur Visualisierung erläutert.

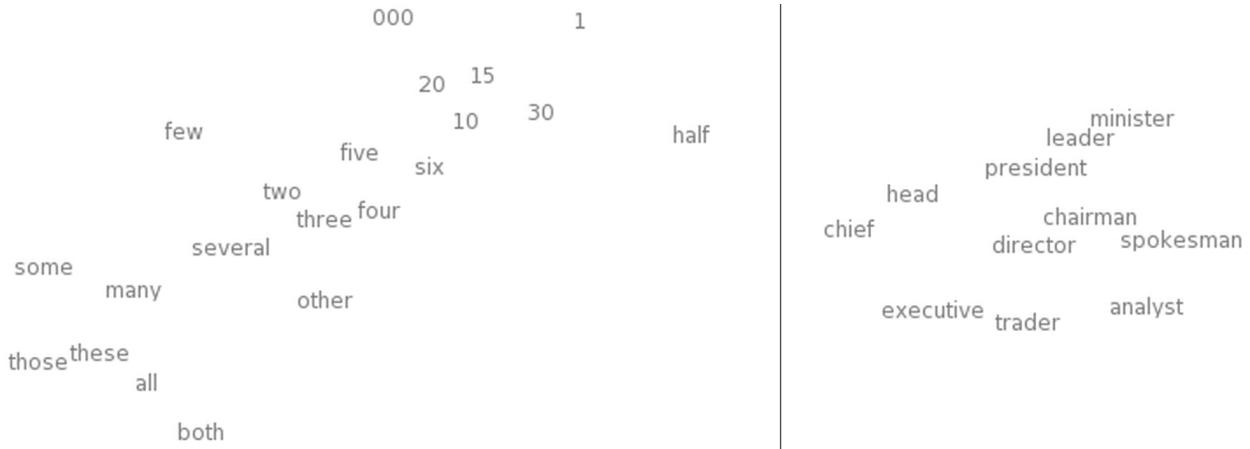
Abbildung 11: Modulares Netzwerk zur Validierung von 5-Grammen mit einer Word embedding Funktion ( $W$ ) und einem neuronalen Netzwerk ( $R$ )



Quelle: Olah (2014)

Die erste Visualisierung bedient sich dem t-SNE<sup>8</sup> Algorithmus um die hoch-dimensionalen Vektoren in einem zweidimensionalen Diagramm darzustellen. Die Abbildung 12 zeigt ein solches Diagramm. Es ist klar zu erkennen, dass ähnliche Wörter nahe zusammen sind (Olah, 2014).

Abbildung 12: t-SNE Darstellung eines Word embeddings



Quelle: Turian, Ratinov und Bengio (2010) in Olah (2014)

Die zweite Visualisierung listet in einer Tabelle (vgl. Tabelle 13) für sechs Wörter die nächsten Embeddings, sprich mit den mathematisch nächsten Vektoren, auf. So werden beispielsweise unter dem Titel *FRANCE* neben *EUROPA* diverse weitere Länder aufgelistet.

Sowohl Abbildung 12 als auch Tabelle 13, zeigen die Stärke von Word embeddings auf. Ähnliche Wörter werden mit ähnlichen Vektoren versehen und so wird eine komplexe Landschaft von zusammengehörigen Wörtern gebildet. Da somit zwei Synonyme ein ähnliches Word embedding aufweisen, verändert sich der Input-Vektor eines nachfolgenden neuronalen Netzwerks durch den Austausch dieser nur geringfügig. Somit muss dieses nachfolgende neuronale Netzwerk nicht für alle Wörter der Welt trainiert werden, sondern kann auf die Generalisierung durch das Word embedding aufbauen (Olah, 2014).

Word embeddings sind zu einem extrem wichtigen Baustein bei der Verarbeitung von natürlichen Texten geworden. Neben Input und Output Repräsentationen bei NLP Tasks können Word embeddings auch Output Repräsentationen in der Objektkennung sein (Olah, 2014).

---

<sup>8</sup>t-Distributed Stochastic Neighbor Embedding (kurz t-SNE) ist eine Technik zur Reduktion von Dimensionen, welche besonders gut für Visualisierungen von hoch-dimensionalen Daten geeignet ist (van der Maaten & Hinton, 2008).

Tabelle 13: Sechs Ausgangswörter mit den ihnen ähnlichsten Word embeddings

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PRUPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DRAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

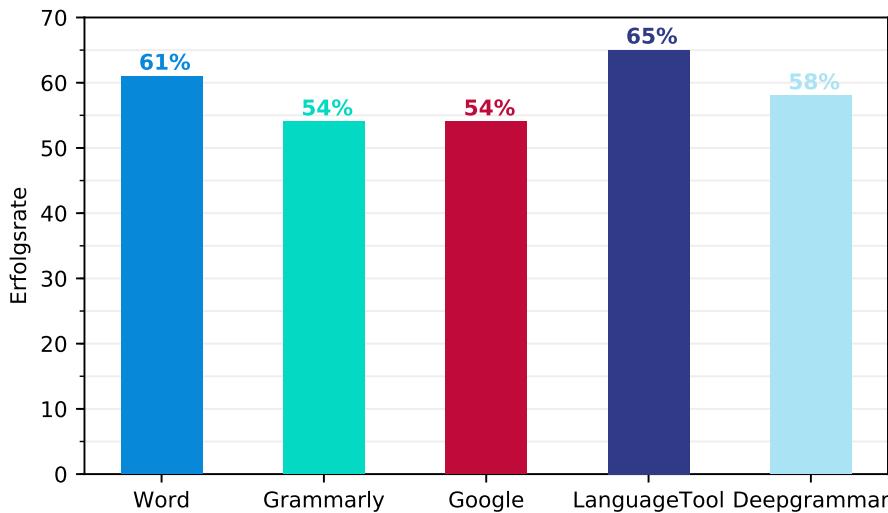
Quelle: Collobert et al. (2011) in Olah (2014)

Für den Prototypen bilden Word embeddings eine wichtige Grundlage. Durch diese Technik können die in den Rechnungen enthaltenen Wörter und Sätze in eine generalisierte, durch neuronale Netzwerke verarbeitbare Form gebracht werden.

### 3.9 Korrektur von Rechtschreibung und Grammatik

Trotz grossem Fortschritt, nicht zuletzt dank der Verwendung von künstlicher Intelligenz, im Bereich der Texterkennung, werden Texte nicht zu 100% korrekt erkannt. So schleichen sich falsch erkannte Buchstaben ein, welche nicht nur Wörter, sondern auch ganze Sätze bedeutungslos machen. Um solche Fehler zu korrigieren, wird auf die Rechtschreibung- und Grammatik-Korrektur zurückgegriffen. Während diverse Korrekturprogramme regelbasierte Software anwenden, wurde auch in diesem Bereich bereits erfolgreich künstliche Intelligenz angewandt. Mit Hilfe des Word embeddings und von LSTM Netzwerken können künstliche Intelligenzen zur Korrektur von Rechtschreibung und Grammatik geschaffen werden. So beschreibt Weiss (2016), wie mit einem einfachen neuronalen Netzwerk, bestehend aus nur 4 LSTM und 4 Dropout Schichten, bereits erfolgreich Rechtschreibfehler korrigiert werden können.

Abbildung 14: Vergleich der Erfolgsrate bei der Prüfung von 418 Textsnippets



Quelle: Mugan (o.D.)

Nicht nur zur Korrektur von Rechtschreibfehlern ist ein neuronales Netzwerk anwendbar, sondern auch zur Grammatikprüfung. So kann unter [deepgrammar.com](https://deepgrammar.com) ein Experiment gefunden werden, bei welchem ein neuronales Netzwerk zur Grammatikprüfung angewendet wird. Die Resultate, welche in der Abbildung 14 zu sehen sind, sind erstaunlich. Obwohl DeepGrammar erst seit einem Jahr existiert und dabei von nur einer Person entwickelt wurde, funktioniert das Netzwerk beinahe so gut wie *Microsoft Word*<sup>9</sup> oder *Language Tool 3.1*<sup>10</sup> und sogar besser als *Grammarly*<sup>11</sup> und *Google Docs*<sup>12</sup> (Mugan, o.D.).

Ein weiterer grosser Vorteil von neuronalen Netzwerken zur Fehlerkorrektur erwähnt Mugan (2018) in einer persönlichen Kommunikation mit dem Autoren dieser Arbeit. Das neuronale Netzwerk kann auf das Domänenspezifische Lexikon trainiert werden. Das Netzwerk kann beispielsweise mit medizinischen Begriffen aus den Rechnungen trainiert werden, so dass die Resultate des in dieser Arbeit entwickelten Prototypen noch besser werden (Mugan, 2018).

---

<sup>9</sup>Microsoft Word ist ein Programm zur Textverarbeitung und Dokumenterstellung von Microsoft (Microsoft Corporation, 2018).

<sup>10</sup>„LanguageTool ist eine Software zur Textprüfung [...]“ (LanguageTool, 2018).

<sup>11</sup>Grammarly verspricht präzise, kontextabhängige Korrekturen von Texten (Grammarly Inc., 2018).

<sup>12</sup>Google Docs ist eine Online-Lösung zur Textverarbeitung von Google (Google LLC, 2018).

## 3.10 Natural Language Processing

Natural Language Processing (kurz NLP) beschreibt die Anwendung von Computern um natürliche Sprache, in Wort und Schrift, zu verstehen und verarbeiten. NLP umfasst Forschungsfelder wie beispielsweise die maschinelle Übersetzung, die Spracherkennung sowie die Informationsextraktion (Chowdhury, 2003).

Für diese Arbeit ist das Themengebiet des Natural Language Processing insofern relevant, als dass es die Grundlage für die Verarbeitung von Rechnungen bildet. Mit Techniken aus diesem Gebiet werden die Inhalte der Rechnungen vom Computer verstanden und verarbeitet.

## 3.11 Informationsextraktion aus natürlichen Texten

Informationsextraktion beschreibt das Themengebiet rund um die Extraktion von strukturierten Informationen aus unstrukturiertem oder halb-strukturiertem Text. In diesem Kapitel werden einige Techniken aus diesem Themengebiet erläutert.

Eine *Regular Expression* (kurz RegEx) ist ein Ausdruck, welcher eine Zeichenkette beschreibt. Diese Ausdrücke funktionieren ähnlich wie arithmetische Ausdrücke. Es werden Operatoren verwendet, um mehrere Ausdrücke zu einem komplexeren Ausdruck zusammenzufassen (Xiao, 2004).

Xiao (2004) beschreibt als einfaches Beispiel den Ausdruck `[a,p]m [0-9]+: [0-9]+` um Zeitangaben wie AM 12:45 zu extrahieren. Dieses Beispiel zeigt einerseits die Einfachheit dieser Technik aber auch die Grenzen. 12:45 AM wird beispielsweise nicht erkannt, da AM hier nach anstelle vor der Uhrzeit steht.

Ein weiterer Nachteil von Regular Expressions ist, dass Kontextinformationen nicht berücksichtigt werden. Folgendes Beispiel von Xiao (2004) zeigt dies auf. Der Ausdruck `[0-9]+` ist zwar in der Lage aus dem Text `100$` die Zahl `100` zu extrahieren, allerdings geht die Information, dass es sich hier um einen Geldbetrag handelt, verloren.

Um eine hohe Präzision bei der Informationsextraktion zu ermöglichen, sollten Regular Expressions also nur mit Vorsicht und in Kombination mit anderen Techniken verwendet werden (Xiao, 2004).

Named Entity Recognition and Classification (kurz NERC oder NER), beschreibt das erkennen und kategorisieren von Entitäten, sprich Wörter oder Wortgruppen aus natürlichen Texten (Nadeau & Sekine, 2007).

Der Begriff Named Entity wurde bei der Formulierung der Aufgabenstellung der sechsten Message Understanding Conference im Jahre 1995 definiert (Borthwick, Sterling, Agichtein & Grishman, 1998). So wurde bereits damals erkannt, dass die Extraktion von Namen von Personen, Organisationen oder Lokationen, nummerischen Ausdrücken, Daten und Prozent-Ausdrücken wichtig ist (Nadeau & Sekine, 2007).

Für die Named Entity Recognition and Classification stehen einige freie Softwarelösungen zur Verfügung. So veröffentlicht beispielsweise Stanford eine Java Implementierung und SpaCy, eine Sammlung von Natural Language Processing Software, beinhaltet eine Implementierung in Python (Stanford NLP Group, o.D.; Explosion AI, o.D.).

Die Anwendung von NERC ist für das Fallbeispiel interessant. Die Erkennung von Namen von Personen ist hilfreich zur Erkennung des Patienten und des Leistungserbringens. Weiter hilft die Erkennung von Daten der Ermittlung des Behandlungsdatums und nicht zuletzt kann durch die Erkennung und Klassifizierung von nummerischen Ausdrücken der Gesamtbetrag sowie die Beträge einzelner Positionen ermittelt werden.

Die letzte Technik, welche in diesem Kapitel erläutert wird, ist das Part of Speech Tagging (kurz PoS-Tagging). Beim PoS-Tagging werden Wörter und Satzzeichen ihren Wortarten (Nomen, Adjektive, etc.) zugewiesen (Xiao, 2004).

Die grösste Herausforderung beim PoS-Tagging sind Wörter welche verschiedenen Wortgruppen zugewiesen werden könnten. Beispielsweise kann das Wort „widerwillig“ im Satz „Sie nannten den Täter widerwillig.“ als Adjektiv oder Adverb aufgefasst werden und somit die Bedeutung des Satzes vollkommen verändern (Volk, o.D.).

## 3.12 Klassifizierung von Bildern

Die Klassifizierung von Bildern ist eine viel behandelte Problemstellung im Bereich der Computer Vision. Dabei geht es darum, ein Bild einer bestimmten Klasse zuzuweisen. Ist auf dem Bild beispielsweise ein Hund zu sehen, so ist es der Klasse Hund zuzuweisen (Goodfellow et al., 2016).

2012 gewann das AlexNet, eines der ersten Convolutional Neural Network basierten Modelle zur Klassifizierung von Bildern, mit einem grossen Abstand die ImageNet Large-Scale Visual Recognition Challenge<sup>13</sup> (kurz ILSVRC). Seither sind neuronale Netzwerke die verbreitetste Methode zur Klassifizierung von Bildern. Diese tiefen neuronalen Netzwerke erreichen heute bessere Resultate als Menschen (Forson, 2017).

Auf dem Machine Learning Blog Towards Data Science beschreiben und vergleichen diverse Autoren verschiedenste Modelle im Bereich der Computer Vision. Der Blog bietet einen guten Überblick über die aktuellen Methoden zur Klassifizierung von Bildern. So werden im Juli 2017 die Residual Networks (kurz ResNet), entwickelt von He, Zhang, Ren und Sun (2015), als die bahnbrechendste Eingabe beim ImageNet LSVRC Wettbewerb der letzten Jahre bezeichnet (Fung, 2017). Im September 2018 beschreibt Tsang (2018) das InceptionV4 Netzwerk von Google, welches vom GoogLeNet abgeleitet und mit den Ideen aus dem ResNet erweitert wurde. Dieses Modell erzielt noch bessere Resultate als das ResNet selbst. Im gleichen Artikel wird auch das Inception-ResNet-V2 vorgestellt, welches im Vergleich zum InceptionV4 Netzwerk schneller trainiert werden kann und zugleich etwas bessere Resultate erzielt.

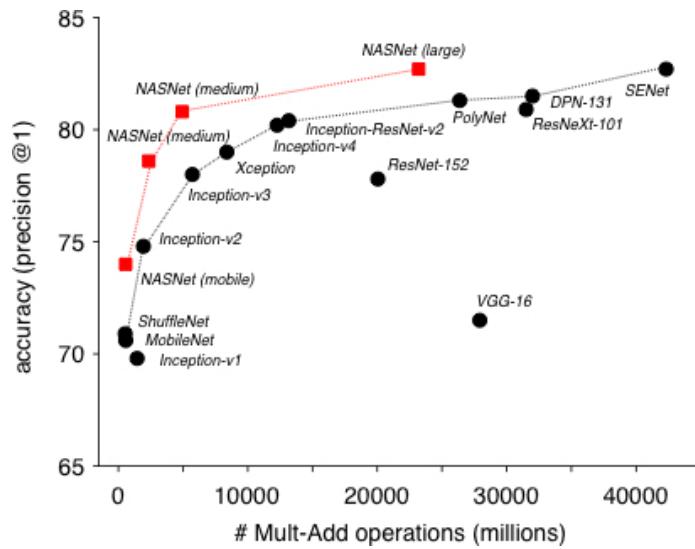
In einem Blog Post auf dem Google AI Blog präsentieren Forscher aus dem Google Brain Team das NASNet. NASNet ist ein Modell zur Klassifizierung von Bildern, welches durch die Anwendung von Machine Learning designed wurde (Zoph, Vasudevan, Shlens & Le, 2017). Unter dem Codename AutoML publiziert das Google Brain Team einen Ansatz, bei welchem ein neuronales Netzwerk ein anderes erstellt - eine künstliche Intelligenz, welche eine neue künstliche Intelligenz schafft. Mit diesem Ansatz kann der sehr aufwendige Design-Prozess zur Schaffung eines neuronalen Netzwerks vereinfacht beziehungsweise automatisiert werden (Le & Zoph, 2017).

---

<sup>13</sup>Die ImageNet Large Scale Visiual Recognition Competition (kurz ILSVRC) ist ein Wettbewerb der ImageNet Organisation, bei welcher hunderte Data Scientists ihre Modelle im Bereich der Computer Vision vergleichen.

Als Teil des gleichen Blog Posts, in welchem das NASNet präsentiert wird, vergleichen die Forscher von Google das Modell mit anderen Netzwerken, wie dem ResNet und dem Inception-ResNet-V2. Der Abbildung 15 ist zu entnehmen, dass das vom Google Brain Team präsentierte NASNet, in der medium Ausprägung, trotz reduzierter Anzahl an benötigten Operationen, sprich reduzierter Komplexität, eine verbesserte Trefferquote als die bisherigen Netzwerke erzielt. In der large Ausprägung kann das NASNet durch eine erhöhte Komplexität eine noch bessere Trefferquote erzielen (Zoph et al., 2017).

Abbildung 15: Vergleich des NASNet mit anderen Netzwerken zur Klassifizierung von Bildern



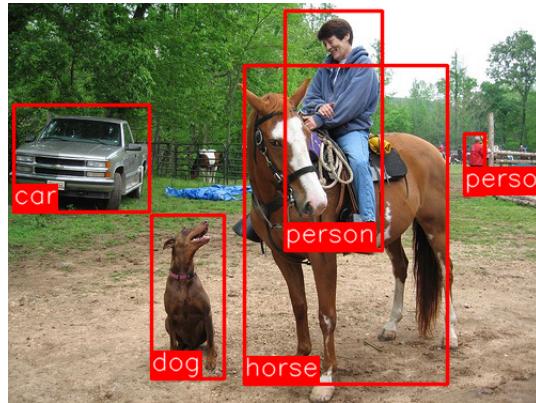
Quelle: Zoph, Vasudevan, Shlens und Le (2017)

### 3.13 Objekterkennung in Bildern

Bei der Objekterkennung in Bildern geht es darum, ein Bild nicht nur einer Klasse zuzuweisen, sondern zu erkennen, wo Objekte auf dem Bild sind. Die einzelnen Objekte werden dann einer Klasse zugewiesen. So könnte ein Bild beispielsweise Personen, Hunde, Pferde und Autos enthalten. Das Modell hat die Aufgabe, alle Objekte dieser Klassen zu umrahmen und zu erkennen, um welche Klasse es sich handelt (Goodfellow et al., 2016).

Das R-CNN Modell ist eines der ersten Modelle, welches die Convolutional Neural Network Architektur nutzt, um nicht nur Bilder zu klassifizieren, sondern auch die Position von Objekten zu erkennen. Die Ausgabe des R-CNN Modells sind mehrere Rechtecke als Umrandungen von Objekten und eine zugehörige Klasse (vgl. Abbildung 16) (Forson, 2017).

Abbildung 16: Resultate aus einem Modell zur Objekterkennung in Bildern



Quelle: Forson (2017)

Die Fast R-CNN und Faster R-CNN Modelle bieten Verbesserung in Hinblick auf die Geschwindigkeit und die Genauigkeit. Das Faster R-CNN Modell ist bis heute eines der exaktesten Modelle zur Objekterkennung in Bildern (Forson, 2017).

Trotz der verbesserten Geschwindigkeit des Faster R-CNN Modells ist mit den auf dem R-CNN Modell basierenden Modellen keine Objekterkennung in Echtzeit möglich. Die Modelle sind hierfür nicht performant genug. Das Single-Shot Multibox Detector (kurz SSD) und das You Only Look Once Modell bieten mit anderen Architekturen eine bessere Performance. Single-Shot bedeutet, dass die Modelle mehrere Objekte in nur einem Feed-Forward Durchgang erkennen können. Durch diese Änderung in der Architektur und dadurch verbesserte Geschwindigkeit, wird an Genauigkeit eingebüsst (Forson, 2017).

## 3.14 Transfer Learning

Transfer Learning beschreibt das Vorgehen, bei welchem das Gelernte aus einer Aufgabe genutzt wird, um das Lernen für eine andere Aufgabe zu vereinfachen (Goodfellow et al., 2016).

Beim Transfer Learning wird ein Modell verwendet, um zwei oder mehr verschiedene Aufgaben zu lösen. Es wird dabei angenommen, dass die Vorhersagen für eine Aufgabe aufgrund ähnlicher Kriterien getroffen werden, wie für jene einer anderen Aufgabe. Dies ist typischerweise dann der Fall, wenn die Eingabewerte eines Modells gleich oder ähnlich sind, sich die Art der gesuchten Ausgabewerte aber unterscheidet (Goodfellow et al., 2016).

Ein Modell wird trainiert, Bilder von Hunden und Katzen zu klassifizieren. Dabei stehen viele Trainingsdaten zur Verfügung. Nun soll ein Modell entwickelt werden, welches Bilder von Ameisen und Wespen klassifiziert. Sind nun für die zweite Aufgabe nur wenige Trainingsdaten verfügbar, so ist es ratsam, das Modell aus der ersten Aufgabe als Grundlage zur Lösung der zweiten Aufgabe zu verwenden. Viele visuelle Repräsentationen teilen grundlegende Eigenschaften wie Formen und Kanten, so kann die Wiederverwendung des trainierten Modells aus der ersten Aufgabe die Trefferquote in der zweiten Aufgabe stark erhöhen (Goodfellow et al., 2016).

Wird ein Modell, welches bereits für eine Aufgabe trainiert wurde, für eine zweite Aufgabe weiter trainiert, spricht man von Fine-tuning (Goodfellow et al., 2016).

Ursprünglich kommt das Transfer Learning aus dem Bereich der Computer Vision. Im Jahr 2018 präsentierte Howard und Ruder (2018) sowie Devlin, Chang, Lee und Toutanova (2018) mit ULMFiT<sup>14</sup> respektive BERT<sup>15</sup> Ansätze, das Transfer Learning auch für Aufgaben im Gebiet des Natural Language Processing anzuwenden. Beide Ansätze versprechen die Reduktion der Menge benötigter Trainingsdatensätze sowie die Verbesserung der resultierenden Modelle.

---

<sup>14</sup>Universal Language Model Fine-tuning, kurz ULMFiT, ist ein Ansatz, bei welchem ein Modell auf einem sehr grossen Datensatz trainiert wird. Dieses Modell kann nun für diverse Aufgaben im Gebiet des Natural Language Processing verwendet werden und verspricht bessere Ergebnisse als Ansätze ohne Transfer Learning (Howard & Ruder, 2018).

<sup>15</sup>Bidirectional Encoder Representations from Transformers, kurz BERT, ist eine neuartige Repräsentation von natürlicher Sprache. Mit diesem auf Transfer Learning basierten Ansatz konnte das Google AI Language Team in diversen NLP Aufgaben Bestresultate erzielen (Devlin et al., 2018).

## 3.15 Messkriterien zur Bewertung eines Systems mit künstlicher Intelligenz

Um ein System objektiv zu bewerten und mit anderen Ansätzen zu vergleichen, bedarf es Metriken. Folgend werden die für diese Arbeit relevanten Metriken erläutert und ihre Anwendungsfälle diskutiert.

### 3.15.1 Trefferquote

Die Trefferquote, englisch Accuracy, ist die Metrik, welche wohl am meisten im Zusammenhang mit der künstlichen Intelligenz zu finden ist (Shung, 2018).

Die Trefferquote gibt Auskunft darüber, wie viele der Vorhersagen eines Systems der Wirklichkeit entsprechen. Die Formell lautet wie folgt (Shung, 2018):

$$\text{Accuracy} = 1 - \frac{\text{False Positive} + \text{False Negative}}{\text{Total Records}} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Records}}$$

Die Confusion Matrix in Abbildung 17 zeigt ein Beispiel mit einer Trefferquote von 0.999, respektive 99.9%. Aus 1000 Datensätzen wurden 998 korrekt als Negativ klassifiziert. Ein Datensatz wurde korrekt als Positiv klassifiziert und ein Datensatz wurde fälschlicherweise als Negativ klassifiziert (Shung, 2018).

Abbildung 17: Beispiel einer Confusion Matrix

		Vorhersage / Klassifizierung	
		Negativ	Positiv
Wirklichkeit	Negativ	998	0
	Positiv	1	1

Quelle: Shung (2018)

Eine Trefferquote von 99.9% ist sehr gut. Das Modell könnte durchaus als sehr erfolgreich betrachtet werden. Diese Betrachtung ist jedoch abhängig vom Anwendungsfall des Modells. Wenn das Modell die Infektion mit einem hoch ansteckenden Virus ermitteln würde, so wäre eine Infektion unentdeckt geblieben. In diesem Fall wäre eine geringere Trefferquote des Modells besser, wenn dafür keine falsche negativ Vorhersagen vorhanden wären (Shung, 2018).

### 3.15.2 Genauigkeit

Die Genauigkeit, englisch Precision, sagt aus, wie viele als positiv vorhergesagten Datensätze wirklich positiv sind. Die Formel dazu lautet (Shung, 2018):

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{True\ Positive}{Total\ Predicted\ Positive}$$

Die Abbildung 18 veranschaulicht die Elemente zur Berechnung der Genauigkeit in einer Confusion Matrix.

Abbildung 18: Elemente zur Berechnung der Genauigkeit in einer Confusion Matrix

		Vorhersage / Klassifizierung	
		Negativ	Positiv
Wirklichkeit	Negativ	True Negative	False Positive
	Positiv	False Negative	True Positive

Quelle: Shung (2018)

Die Genauigkeit bietet sich immer dann als gute Metrik an, wenn die Kosten eines False Positive hoch sind. Ein Beispiel dafür ist ein Spamfilter. Markiert der Spamfilter ein E-Mail fälschlicherweise als kein Spam, so ist dies kein Problem. Der Anwender kann das E-Mail manuell als Spam markieren. Markiert der Spamfilter jedoch ein E-Mail fälschlicherweise als Spam, so ist die Wahrscheinlichkeit hoch, dass der Anwender das E-Mail nie lesen wird (Shung, 2018).

### 3.15.3 Sensitivität

Die Sensitivität, englisch Recall, sagt aus, wie viele wirklich positiven Datensätze als positiv vorhergesagt werden. Die Formel zur Berechnung der Sensitivität lautet (Shung, 2018):

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{True\ Positive}{Total\ Actual\ Positive}$$

Die Abbildung 19 veranschaulicht die Elemente zur Berechnung der Sensitivität in einer Confusion Matrix.

Abbildung 19: Elemente zur Berechnung der Sensitivität in einer Confusion Matrix

		Vorhersage / Klassifizierung	
		Negativ	Positiv
Wirklichkeit	Negativ	True Negative	False Positive
	Positiv	False Negative	True Positive

Quelle: Shung (2018)

Die Sensitivität ist immer dann relevant, wenn die Auswirkungen durch ein False Negative gross sind. Ein Beispiel dafür ist die Klassifizierung von Banktransaktionen mit den Klassen Betrug und kein Betrug. Wird eine reguläre Transaktion als Betrug markiert, so kann dies abgeklärt und korrigiert werden. Wird jedoch eine betrügerische Transaktion nicht als solche erkannt, so sind die finanziellen Auswirkungen womöglich gross (Shung, 2018).

#### 3.15.4 F-Mass

Das F-Mass, englisch  $F_1$ -Score, vereint die Genauigkeit und die Sensitivität. Im Gegensatz zur Trefferquote lässt sich der Wert aber nicht durch ein Übermass an True Negatives beeinflussen, denn sie finden keine Beachtung bei der Berechnung. In den meisten Anwendungsfällen sind True Negatives nicht relevant. False Negatives und False Positives sind meist die Verursacher von hohen Kosten. Das F-Mass wird mit folgender Formel berechnet (Shung, 2018):

$$F_1 = 2 * \frac{\text{Genauigkeit} * \text{Sensitivität}}{\text{Genauigkeit} + \text{Sensitivität}}$$

Das F-Mass ist eine gute Verbindung der Genauigkeit und der Sensitivität. Aufgrund der Vernachlässigung der True Negatives ist das F-Mass besonders bei ungleichmässig verteilten Klassen, sprich übermäßig vielen True Negatives, sinnvoll (Shung, 2018).

### 3.15.5 Loss und Loss-Funktion

Das sogenannte Loss, auch Kosten oder Fehler, zeigt, wie sehr eine Vorhersage von der Realität abweicht. Um das Loss zu berechnen, kommt die Loss-Funktion, auch Kosten- oder Fehler-Funktion, zur Anwendung. Wie genau die Loss-Funktion definiert wird, hängt von der Problemstellung ab. Bei einer binären Klassifikation ist beispielsweise die Cross-Entropy-Loss-Function oder die Log-Loss-Function gängig (Godoy, 2018).

Während dem Training ist das Loss, die Differenz zwischen Vorhersage des Modells und den Trainingsdaten, die zu optimierende Grösse. Die Wahl der Loss-Funktion kann einen erheblichen Einfluss auf das Modell haben. Die Loss-Funktion wird während dem Training meist durch Regularisierungstechniken erweitert, um ein Overfitting zu verhindern (Goodfellow et al., 2016).

### 3.15.6 Intersection over Union

Um die Trefferquote, Genauigkeit und die Sensitivität zu berechnen, muss bekannt sein, ob eine Vorhersage eines Modells korrekt war oder nicht. Das bedeutet, es muss bekannt sein, ob es sich um einen True Positive, False Positive, True Negative oder False Negative handelt.

Bei der Objekterkennung in Bildern ist dies nicht immer eindeutig zu sagen. Ein Modell zur Objekterkennung sagt eine Position eines Objektes, als umschliessendes Rechteck, sowie dessen Klasse vorher. Die Korrektheit der Klasse kann klar beurteilt werden. Entweder die Vorhersage stimmt mit der Erwartung überein oder nicht. Die Beurteilung der Korrektheit der vorhergesagten Position ist dagegen nicht eindeutig, wie dies Abbildung 20 verdeutlicht. Die Vorhersage des Modells trifft nicht exakt die Erwartung, doch deshalb ist die Vorhersage nicht falsch. Eine exakte Übereinstimmung des vorhergesagten und des erwarteten Rechtecks ist unwahrscheinlich. Aus diesem Grund ist eine Metrik notwendig, die solche Ungenauigkeiten zulässt (Rosebrock, 2016).

Intersection over Union, kurz IoU, ist ein solches Mass. Es stellt, wie es der Name bereits sagt, die Überlappung des vorhergesagten mit dem erwarteten Rechteck in das Verhältnis zur Vereinigung dieser (Rosebrock, 2016):

$$IoU = \frac{\text{Fläche der Überlappung}}{\text{Fläche der Vereinigung}}$$


Abbildung 20: Beispiel einer tatsächlichen und vorhergesagten Position eines Objekts

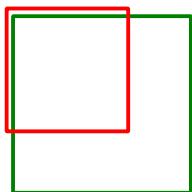


Quelle: Rosebrock (2016)

Die Beispiele aus Abbildung 21 verdeutlichen, dass das Mass gegen 1 tendiert, je exakter das vorhergesagte mit dem erwarteten Rechteck übereinstimmt. Es kann damit beurteilt werden, wie genau die Vorhersage ist. Wird nun noch ein Schwellenwert definiert, ab welchem eine Vorhersage als Korrekt angesehen wird, so kann die Korrektheit der Vorhersage ermittelt werden. Die Wahl dieses Schwellenwerts ist sehr unterschiedlich. In Wettbewerben zur Objekterkennung sind Schwellenwerte zwischen 0.5 und 0.75 üblich. Auch wird teilweise ein Durchschnitt über mehrere Schwellenwerte verwendet (Rosebrock, 2016).

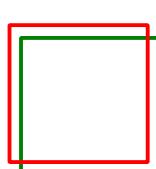
Abbildung 21: Beispiele des Intersection over Union Mass

$$IoU = 0.4034$$



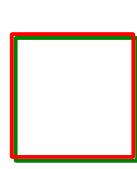
Schlecht

$$IoU = 0.7330$$



Gut

$$IoU = 0.9264$$



Sehr gut

Quelle: Rosebrock (2016)

### 3.15.7 Average Precision

Average Precision, kurz AP, ist eine Metrik, welche oft bei der Bewertung von Modellen zur Objekterkennung zur Anwendung kommt. Als Average Precision wird der Durchschnitt aller Genauigkeiten bei jeder Sensitivität verstanden. Um dies genauer zu erläutern wird im Folgenden ein Beispiel zur Berechnung aufgeführt (Hui, 2018a).

Ein Modell hat die Aufgabe, Katzen auf Bildern zu erkennen. Auf fünf von zehn Bildern ist eine Katze zu sehen. Das heisst, es sind insgesamt fünf True Positive Beispiele vorhanden. Die Tabelle 22 zeigt mögliche Ergebnisse des Modells. Die Ergebnisse sind sortiert nach der Konfidenz des Modells, richtig zu liegen. Pro Vorhersage wird nun die Genauigkeit sowie die Sensitivität berechnet. Dabei steigt die Sensitivität bei jeder korrekten Vorhersage. Die Genauigkeit steigt bei jeder korrekten Vorhersage und sinkt bei jeder falschen (Hui, 2018a).

Tabelle 22: Beispiel der Berechnung der Genauigkeit und Sensitivität

Rang	Korrekte Vorhersage	Genauigkeit	Sensitivität
1	Ja	1.0	0.2
2	Ja	1.0	0.4
3	Nein	0.67	0.4
4	Nein	0.5	0.4
5	Nein	0.4	0.4
6	Ja	0.5	0.6
7	Ja	0.57	0.8
8	Nein	0.5	0.8
9	Nein	0.44	0.8
10	Ja	0.5	1.0

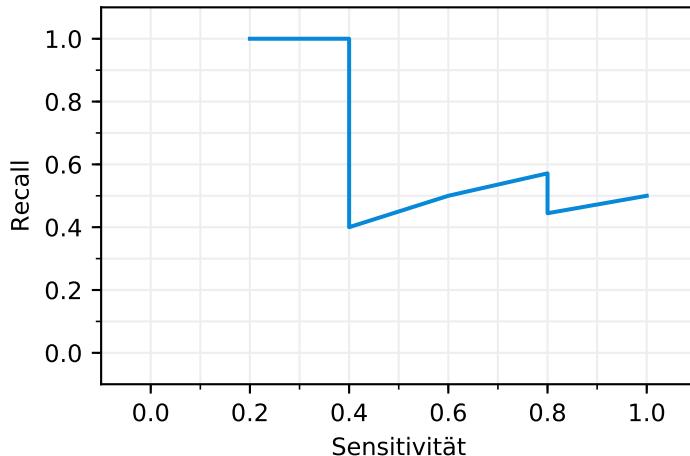
Quelle: Hui (2018a)

Die Abbildung 23 zeigt die sogenannte Precision-Recall-Curve (kurz PR-Curve) für das Beispiel aus Tabelle 22. Die PR-Curve zeigt die Genauigkeit bei jeder Sensitivität. Die Sensitivität ist aufgrund ihrer Definition immer monoton steigend. Die Genauigkeit steigt beziehungsweise sinkt bei korrekten respektive falschen Vorhersagen. Auf der Abbildung ist die daraus resultierende Zickzacklinie zu sehen (Hui, 2018a).

Wie erwähnt, wird als Average Precision der Durchschnitt aller Genauigkeiten bei jeder Sensitivität verstanden. Die Average Precision wird generell als die Fläche unter der PR-Curve definiert. Die Formel zur Berechnung der Average Precision lautet (Hui, 2018a):

$$AP = \int_0^1 p(r)dr$$

Abbildung 23: Beispiel einer PR-Curve

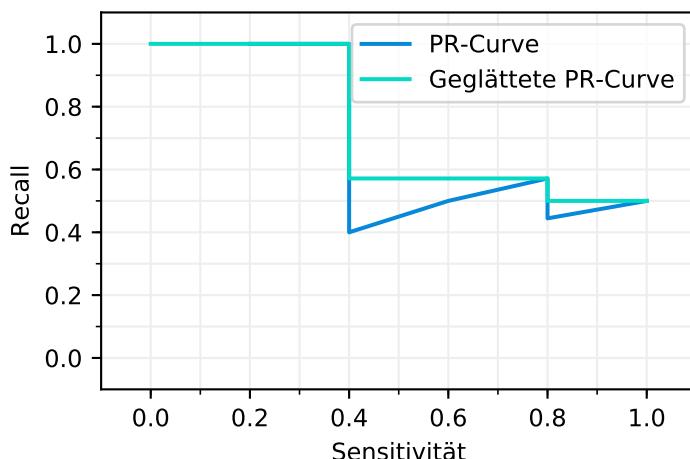


Quelle: Hui (2018a)

Da sowohl die Genauigkeit als auch die Sensitivität zwischen 0 und 1 liegt, fällt auch die Average Precision zwischen 0 und 1 (Hui, 2018a).

Um die Berechnung der Average Precision zu vereinfachen, wird die Zickzacklinie oftmals geglättet. Dafür wird zu jeder Sensitivität jeweils die maximale Genauigkeit aller grösseren Sensitivitäten gewählt (Hui, 2018a). Die grüne Linie in Abbildung 24 veranschaulicht diese Glättung.

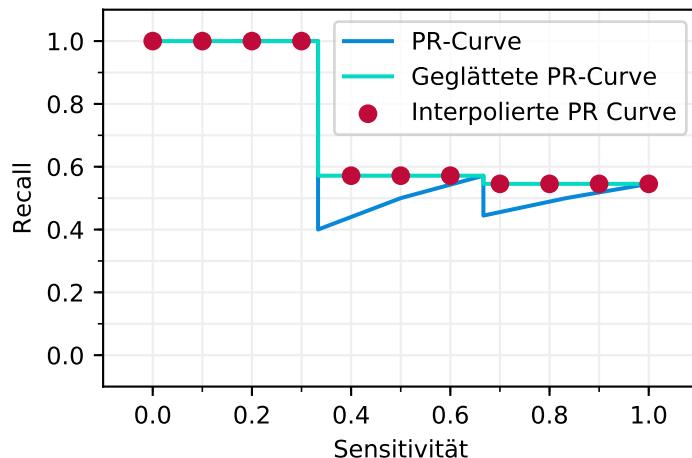
Abbildung 24: Beispiel zur Glättung einer PR-Curve



Quelle: Hui (2018a)

Neben dieser Glättung kommt in gewissen Fällen auch eine Interpolation zur Anwendung. Bis 2008 verwendete das PASCAL VOC Dataset<sup>16</sup> zur Berechnung der Average Precision eine Interpolation mit elf Punkten. Das bedeutet, dass die Genauigkeit bei allen Sensitivitäts-Werten mit dem Faktor 0.1 berechnet wird. Beim COCO Dataset<sup>17</sup> wird eine Interpolation von 101 Punkten verwendet. Im neuen PASCAL VOC Dataset wird auf eine Interpolation verzichtet und die Average Precision wird als die Fläche unter der geglätteten PR-Curve berechnet. In der Abbildung 25 ist ersichtlich, dass je nach Art der Berechnung unterschiedliche Werte resultieren. Durch die Interpolation können Ungenauigkeiten entstehen, wenn die Abnahme der Genauigkeit nicht genau auf einen der zur Berechnung verwendeten Punkte fällt. Beim Vergleich von Modellen muss darauf geachtet werden, dass die Modelle nicht aufgrund unterschiedlich berechneter Average Precisions beurteilt werden. Ansonsten kann ein falscher Eindruck entstehen (Hui, 2018a).

Abbildung 25: Interpolation einer geglätteten PR-Curve.



Quelle: Hui (2018a)

<sup>16</sup>Die PASCAL Visual Object Classes Datasets sind ein annotierte Datensätze zur Objekterkennung auf Bildern, welche von 2005 bis 2012 für die PASCAL VOC Challenge ausgegeben wurden.

<sup>17</sup>Das COCO (Common Objects in Context) Dataset ist ein sehr grosses Dataset, welches für diverse Wettbewerbe im Bereich der Objekterkennung verwendet wird.

### 3.15.8 Mean Average Precision

Als Mean Average Precision (kurz mAP) wird der Durchschnitt über mehrere Average Precisions bezeichnet. Welche Average Precisions dabei gemeint sind, ist von Anwendung zu Anwendung unterschiedlich. In gewissen Fällen wird einfach nur von der Average Precision gesprochen, obwohl eigentlich ein Durchschnitt über mehrere Average Precisions gemeint ist. Es wird davon ausgegangen, dass dies im Kontext klar ist. Im Folgenden werden drei Varianten zur Bildung der Mean Average Precision erläutert (Hui, 2018a).

Im COCO Dataset wird der Durchschnitt der Average Precisions für unterschiedliche Schwellenwerte für das Intersection over Union Verfahren als Mean Average Precision bezeichnet. Konkret werden alle Average Precisions für die IoU Schwellenwerte in Schritten von 0.05 zwischen 0.5 und 0.95 gemittelt. Dies wird dabei als  $mAP[0.5:0.05:0.95]$  oder kürzer  $mAP[0.5:0.95]$  bezeichnet. Durch dieses Verfahren werden Modelle mit genaueren Vorhersagen, sprich grösserem IoU Wert, besser bewertet (Hui, 2018a).

In anderen Fällen wird die Mean Average Precision als der Durchschnitt der Average Precisions über alle zu identifizierenden Klassen definiert. Sollen in einer Aufgabe Hunde und Katzen auf Bildern erkannt werden, so kann für beide Klassen eine Average Precision ermittelt werden. Ein Modell könnte eine Average Precision für Hunde von 0.9 und eine für Katzen von 0.3 aufweisen. Um dieses Modell nun mit einem anderen Modell zu vergleichen, wird die Mean Average Precision über alle Klassen berechnet. Dafür kommt folgende Formel zur Anwendung (Hui, 2018a):

$$mAP = \frac{\sum_{i=0}^n AP_i}{n} = \frac{AP_{Hund} + AP_{Katze}}{2} = 0.6$$

Die beiden Methoden werden oft kombiniert, um die Mean Average Precision über mehrere Intersection over Union Schwellenwerte und über mehrere Klassen zu berechnen. Dies ermöglicht, verschiedene Modelle mit nur einer Zahl zu vergleichen (Hui, 2018a).

## 3.16 Fehleranalyse

Als Fehleranalyse wird die Analyse der falschen Vorhersagen eines Modells bezeichnet. Im Falle eines Klassifizierungsmodells werden die falsch klassifizierten Datensätze betrachtet und auf die Ursache der falschen Klassifizierung untersucht. Das Ziel der Fehleranalyse ist es, Optimierungspotential zu finden, anhand welchem das bestehende Modell verbessert werden kann (Ng, 2018).

Wird beispielsweise ein Modell zur Klassifizierung von Bildern in die Klassen Hund und Katze mit einer Trefferquote von 90% untersucht, so könnte eine Erkenntnis sein, dass nur 5% aller falsch klassifizierten Bilder der Klasse Hund angehören. Durch Optimierungen des Modells zur besseren Erkennung von Hunden könnte nur 5% des Fehlers reduziert, sprich 0.5% an Trefferquote gewonnen werden. Würden 50% der falsch Klassifizierten Bilder der Klasse Hund angehören, würde eine solche Optimierung einen grösseren Einfluss auf die Trefferquote haben. Es könnte im Optimalfall eine Steigerung der Trefferquote von 5% (50% der Fehlerquote von 10%) erreicht werden (Ng, 2018).

Die Fehleranalyse ermöglicht die Ursachen eines falschen Resultates zu finden. Sie kann somit Auskunft darüber geben, wie das untersuchte Modell optimiert respektive weiterentwickelt werden soll (Ng, 2018).

## 3.17 Design eines Systems mit künstlicher Intelligenz

Die vorherigen Kapitel haben einen Einblick in einige der Grundbausteine von Machine Learning Modellen gegeben. Eine grosse Herausforderung ist nun, diese so anzuwenden, dass eine Problemstellung möglichst optimal gelöst werden kann. Dieses Kapitel soll einen Überblick darüber geben, wie ein Machine Learning Modell entwickelt werden kann.

Der erste und einer der wichtigsten Schritte ist, die **Beschreibung der Problemstellung**. Dabei gilt es zu klären, was das Ziel ist respektive was vorhergesagt werden soll. Es muss geklärt werden, wie genau die Ausgabe des Modells aussehen soll. Weiter muss geklärt werden, welche Daten als Eingabewerte notwendig und ob diese verfügbar sind. Weiter gilt es zu definieren, anhand welcher Metrik das Modell gemessen wird (Román Aragay, 2018).

Der nächste Schritt ist die **Beschaffung der identifizierten Daten**. Für gewisse Problemstellungen sind bereits strukturierte Daten vorhanden. Bei anderen müssen diese mit aufwendigen Techniken, wie beispielsweise dem Web-Scraping, dem automatisierten extrahieren von Informationen aus Webseiten, beschafft werden (Román Aragay, 2018).

Im dritten Schritt muss eine **Zielmetrik** bestimmt werden. Die Wahl der Metrik ist wichtig, damit das Ziel bekannt ist und das Modell auf die Erreichung dieses optimiert werden kann (Román Aragay, 2018).

Nachdem das Ziel definiert ist, müssen die **Eingabedaten aufbereitet** werden. Dabei muss ein Weg gefunden werden, mit potentiell fehlenden oder unvollständigen Daten umzugehen. Alle Daten müssen in eine Form gebracht werden, welche das Modell verstehen kann. So müssen beispielsweise ordinale und nominale Daten in Ganzzahlen umgewandelt respektive codiert werden. Da die meisten Modelle mit gleich skalierten Daten am besten umgehen können, werden die Daten auf die Skala [0 : 1] gebracht (Román Aragay, 2018).

Je nach Problematik ist es Ratsam eine **einfache Vorhersage**, beispielsweise basierenden auf k-Nearest-Neighbor<sup>18</sup> oder Naïve Bayes<sup>19</sup>, zu erstellen. Eine solche Vorhersage ermöglicht die Eingabedaten auf ihre Eignung zur Vorhersage des gesuchten Resultats zu prüfen. Das Resultat dieser einfachen Vorhersage kann als Grundlage zur Bewertung des Machine Learning Modells verwendet werden (Ariño de la Rubia, 2017).

Nachdem die Rahmenbedingungen geschaffen wurden, geht es an die **Entwicklung des Modells**. Dabei ist Cross Validation, der Vergleich von bestehenden Modellen, und das anschliessende Optimieren des besten Kandidaten, ein oft angewendetes Vorgehen. Bei der Cross Validation werden verschiedene Modelle auf den Eingabedaten angewendet und anhand der Zielmetrik bewertet. Das Modell, welches die dem besten Resultat erzielt, wird anschliessend optimiert. Diese Optimierung besteht aus der Anpassung der Hyperparameter (Román Aragay, 2018).

#### 3.17.1 Design eines neuronalen Netzwerkes

Um eine Problemstellung mit einem neuronalen Netzwerk zu lösen, muss erst ein solches entworfen werden. Für das Design eines neuronalen Netzwerks gibt es keine klaren Regeln, vielmehr ist es eine Kunst. Der Input sowie der Output Layer sind oftmals durch die Problemstellung gegeben. Die Eigenschaften der zu analysierenden Problemstellung, sogenannte Features, definieren die Form des Input Layers. Bei gewissen Problemstellungen ist die Repräsentation dieser Eigenschaften klar. Zahlen werden auf eine einheitliche Skala skaliert und jede Zahl wird durch ein Neuron im Input Layer repräsentiert. Andere Arten von Eigenschaften, beispielsweise Wörter oder Sätze, können auf verschiedene Arten abgebildet werden. Im Fall von Wörtern und Sätzen gibt das gewählte Word embedding die Form des Input Layer vor. Der Output Layer ist durch die gesuchte Lösung gegeben. Im Fall eines Klassifizierungsproblems wird meist eine One-Hot encoded Vektor als Output Layer verwendet. Dabei werden so viele Neuronen wie Anzahl möglicher Klassen angelegt. Das Neuron, welches die grösste Aktivierung vorweist, bestimmt die vorhergesagte Klasse (Nielsen, 2018).

---

<sup>18</sup>k-Nearest-Neighbor ist ein Algorithmus zur Klassifizierung eines Datensatzes aufgrund seiner k-nächsten Nachbarn, sprich k-ähnlichsten Datensätzen.

<sup>19</sup>Naïve Bayes ist ein Algorithmus zur Klassifizierung, welcher Datensätze aufgrund einer Kostenfunktion klassifiziert. Ein Datensatz wird jener Klasse zugewiesen, bei welcher die kleinsten Kosten entstehen.

Die Wahl der Anzahl, Art und Grösse der Hidden Layer ist die wahre Kunst beim Design eines neuronalen Netzwerks. Diese Entscheidungen sind meist eine Abwägung zwischen Trainingsaufwand und Genauigkeit des Modells. Für einfache Problemstellungen können viele versteckte Schichten schnell zum Overfitting führen. Bei komplexeren Problemstellungen sind dafür viele versteckte Schichten notwendig. Ein Modell mit nur einem einzigen Fully Connected Hidden Layer ist in der Lage eine handschriftliche Zahl zu erkennen. Für eine komplette OCR Lösung, welche Wörter, Sätze und ganze Paragraphen erkennt, ist ein tiefes Modell mit komplexen Schichten, wie LSTM Schichten, notwendig (Nielsen, 2018).

## 3.18 End-to-end Entwicklung von künstlicher Intelligenz

Dank des grossen Fortschritts im Bereich der künstlichen Intelligenz, können mit einfachen Mitteln bereits Modelle erstellt und trainiert werden. Trotz dieses grossen Fortschrittes ist die end-to-end Entwicklung und der Betrieb eines solchen Modells noch immer sehr aufwendig. Aus diesem Grund haben Bailis, Olukotun, Ré und Zaharia (2017) das fünf Jahre dauernde DAWN Forschungsprojekt an der Stanford Universität ins Leben gerufen. DAWN steht für Data Analytics for What's Next. Das Forschungsprojekt hat nicht zum Ziel, die Algorithmen im Bereich der künstlichen Intelligenz zu verbessern, sondern diese einfacher nutzbar zu machen. Damit in Zukunft keine grossen, kostenintensiven Teams von Statistikerinnen und Statistikern und Ingenieurinnen und Ingenieuren mehr benötigt werden, um erfolgreich eine künstliche Intelligenz zu entwickeln und betreiben, sind starke Verbesserungen im Bereich der Nutzbarkeit der Techniken rund um die künstliche Intelligenz notwendig (Bailis et al., 2017).

Der Versuch, kleine Teams von Fachspezialistinnen und Fachspezialisten zu befähigen, künstliche Intelligenzen zu entwickeln und zu betreiben, vergleicht die Forschungsgruppe mit den Entwicklungen im Bereich der Suchmaschinen und Datenbanken. Während früher Suchmaschinen extrem komplex zu integrieren waren, kann eine solche heutzutage im Handumdrehen in eine Applikation integriert werden. Dabei liefert sie out-of-the-box bereits gute Ergebnisse. In den 1970er Jahren wurde durch die relationalen Datenbanken eine ähnliche Entwicklung angestoßen. Vor den relationalen Datenbanken war der Betrieb und die Integration einer Datenbank komplex und aufwendig. Heute betreiben Unternehmen ohne grossen Aufwand eine vielzahl von Applikationen, welche auf Datenbanken basieren (Bailis et al., 2017).

Das Forschungsprojekt hat zum Ziel, in den nächsten Jahren einen Stack zu entwickeln, welcher von der Hardware bis zu den Interfaces die end-to-end Entwicklung einer künstlichen Intelligenz vereinfacht. Dabei legt das Projekt besonders viel Wert auf die Aufbereitung von Daten, Selektion und Extraktion von Eigenschaften dieser Daten (sogenannte Features) sowie den Produktiven Betrieb der künstlichen Intelligenz (Bailis et al., 2017).

## **4 | Automatisierung der Rechnungseinreichung der AXA Gesundheitsvorsorge**

In diesem Kapitel wird das Fallbeispiel erarbeitet, welches identifizieren soll, ob künstliche Intelligenz mit kleinem Budget und wenigen Ressourcen zur Automatisierung von Geschäftsprozessen angewendet werden kann.

Im Folgenden wird eine Einführung in das Fallbeispiel gegeben und der relevante Prozess im Detail erläutert. Es werden Anforderungen zur Automatisierung des relevanten Prozessschrittes und somit des gesamten Prozesses definiert.

Das Vorgehen und die Methodik dieses explorativen Teils werden im Kapitel 4.3 erläutert. Anhand dieses Vorgehen werden zwei für die Automatisierung des Prozessschrittes relevante Aspekte betrachtet. Zu beiden Aspekten werden jeweils Experimente durchgeführt, mit welchen die Machbarkeit der Automatisierung identifiziert werden soll.

Zu jedem Experiment werden die Ergebnisse diskutiert, mögliche Fehlerquellen analysiert und Optimierungspotential aufgezeigt.

### **4.1 Einführung in das Fallbeispiel**

In der Schweiz beliefen sich die Kosten für das Gesundheitswesen im Jahr 2015 auf 77.8 Milliarden Franken. Über 35% dieser Kosten wurden durch die obligatorische Krankenversicherung gedeckt. Weitere knapp 7% wurden von den Zusatzversicherungen übernommen. Die Krankenversicherer finanzierten also mit knapp 42% einen beträchtlichen Teil des Gesundheitswesens in der Schweiz. Die übrigen 58% werden durch den Staat, andere Sozial- oder Privatversicherungen sowie von selbstzahlenden Patienten getragen (BfS, 2018).

Die Kosten des Gesundheitswesen steigen stetig an, so zeigen die Zahlen vom Jahr 2016 bereits Kosten von über 80 Milliarden Franken auf (BfS, 2018). Auch in den folgenden Jahren sollen die Kosten weiter steigen. Kirchgässner (2009) begründet diesen Anstieg unter anderem mit der Veränderung der Altersstruktur, dem steigenden Wohlstand sowie den neuen Möglichkeiten in der Diagnose und Behandlung durch technischen Fortschritt.

Die Kosten, welche die Krankenversicherer tragen, werden mit einem von zwei Systemen, Tiers payant oder Tiers garant, vergütet (vgl. Tabelle 26) (EDI, 2017).

Tabelle 26: Vergütungsmodelle bei den Schweizer Krankenversicherern

Tiers payant	Kosten werden vom Leistungserbringer direkt dem Krankenversicherer in Rechnung gestellt.
Tiers garant	Kosten werden vom Leistungserbringer dem Patienten in Rechnung gestellt, welcher die Rechnung dem Krankenversicherer zur Rückvergütung weiterleitet.

Beim System Tiers payant belastet der Leistungserbringer (bspw. Arzt oder Apotheke) die Kosten direkt dem Krankenversicherer. Dies geschieht, indem der Patient mit seiner Versichertenkarte bezahlt. Anhand der Versichertenkarte, welche vom Krankenversicherer ausgestellt wird, können Deckungen für den Patienten überprüft sowie die Rechnung direkt an den Krankenversicherer übermittelt werden. In diesem Fall wird die Rechnung bereits in digitaler, strukturierter Form übermittelt (EDI, 2017).

Werden Kosten, welche über Tiers payant abgerechnet wurden, nicht vom Krankenversicherer getragen, weil beispielsweise ein Selbstbehalt vereinbart wurde, die Franchise noch nicht aufgebraucht ist oder der Patient für diese Behandlung nicht versichert ist, verrechnet der Krankenversicherer die Kosten dem Patienten weiter (EDI, 2017).

Das System Tiers payant wird häufig in Apotheken, beim Kauf von Medikamenten mit oder ohne ärztlichem Rezept, sowie bei allen stationären Behandlungen, gemäss KVG (Krankenversicherungsgesetz, Art. 42 Abs. 2 ), verwendet (EDI, 2017).

Die Verarbeitung von Rechnungen, welche über das System Tiers payant abgerechnet werden, kann der Krankenversicherer, aufgrund der digitalen, strukturierten Daten, automatisiert gestalten (BAG, 2016).

Im Fall von Tiers garant stellt der Leistungserbringer die Rechnung direkt dem Patienten aus, welcher diese dann seinem Krankenversicherer zur Rückerstattung weiterleitet. Die Rechnung kann bei allen Krankenversicherern per Post oder auch digital, im Kundenportal oder in der App, eingereicht werden (EDI, 2017).

## 4.1 Einführung in das Fallbeispiel

---

Rechnungen, welche per Post oder digital beim Krankenversicherer zur Rückvergütung eingehen, erreichen diesen in verschiedenen Formen und unterschiedlichster Qualität.

Während einige Rechnungen nach dem TARMED Standard für Rückforderungsbelege strukturiert sind, sind andere formlos. Die Bandbreite dieser formlosen Rechnungen ist gross. Von handgeschriebenen Rechnungen eines örtlichen Leistungserbringer bis hin zu strukturierten Rechnungen von grösseren Fitnessketten.

Bei der Einreichung per Post kann die Qualität durch Kaffeeflecken, Verbleichung der Belege oder sonstige Beeinträchtigungen gemindert werden. Der Krankenversicherer kann aber einiges dazu beitragen, die Rechnung in hoher Qualität einzulesen. So kann er beispielsweise hochauflösende Scanner und eine optimale Beleuchtung einsetzen.

Problematischer sind Rechnungen, welche von Kundinnen und Kunden digital, sprich als Foto, an den Krankenversicherer übermittelt werden. Wird ein Foto einer Rechnung über das Kundenportal eingereicht, so hat der Krankenversicherer nur noch sehr wenig Einfluss auf die Qualität der Aufnahme. Schlechte Belichtung, kleine Auflösung und abgeschnittene Rechnungen sind nur wenige der Probleme, mit welchen der Krankenversicherer zu kämpfen hat.

Egal wie und in welcher Qualität eine Rechnung einen Krankenversicherer erreicht hat, muss dieser die Rechnung in eine elektronische, strukturierte Form bringen, damit diese dann durch ein Regelwerk verarbeitet werden kann. Dieser Vorgang wird durch verschiedenste Techniken aus dem Bereich der Texterkennung und der Informationsextraktion ermöglicht.

Die AXA, eine internationale Versicherungsgesellschaft, sieht sich, genau wie alle anderen Krankenversicherer, ebenfalls vor der Herausforderung der Indexierung von Rechnungen. Im Jahr 2017 lancierte die AXA eine Zusatzversicherung in der Branche der Gesundheitsvorsorge auf dem Schweizer Markt. Neben den Zusatzversicherungen selbst bietet die AXA ihren Kundinnen und Kunden einen Rechnungs-Weiterleitungs-Service an. Das bedeutet, alle Rechnungen können der AXA gesendet werden, auch wenn diese die Grundversicherung betreffen. Rechnungen beziehungsweise Rechnungspositionen, welche die Zusatzversicherung betreffen, werden von der AXA vergütet. Rechnungspositionen, welche die Grundversicherung betreffen, werden zur Vergütung an den Grundversicherer weitergeleitet (Finanzen.ch, 2017).

Gemäss einem Beitrag auf Finanzen.ch (2017) ist es das Ziel der AXA, bis im Jahr 2020 insgesamt 100'000 Versicherte für die Gesundheitsvorsorge zu gewinnen. Aus dem Geschäftsbericht der CSS Gruppe für das Jahr 2017 geht hervor, dass für knapp 1.7 Millionen Versicherte 16 Millionen Rechnungen geprüft wurden (CSS Gruppe, 2018). Die Hochrechnung der durchschnittlich 9.5 Rechnungen pro Versicherten der CSS Gruppe auf die Zielgrösse von 100'000 Versicherten der AXA zeigt, dass im Jahr 2020 knapp 1 Million Rechnungen verarbeitet werden müssen. Damit diese Menge an Rechnungen effizient verarbeitet werden kann, ist es für die AXA ein Anliegen, den Prozessschritt der Indexierung möglichst automatisiert zu gestalten.

Um die Verwaltungskosten der Zusatzversicherung aufzuzeigen wird wiederum der Geschäftsbericht der CSS Gruppe für das Jahr 2017 herangezogen. Der Kostensatz, welcher den Anteil der administrativen Kosten am Umsatz misst, betrug für die Grundversicherung lediglich 4%. Im Geschäftsbereich der Zusatzversicherungen liegt dieser Kostensatz allerdings viel höher, nämlich bei 21% (CSS Gruppe, 2018). Welcher Anteil an diesen Kosten nun der Prüfung respektive der Indexierung eingehender Rechnungen zuzuschreiben ist, bleibt ein Betriebsgeheimnis. Da die Rechnungen, welche die Zusatzversicherungen betreffen, aufgrund der unterschiedlichen Leistungserbringer (Alternativmedizin, Fitness, Sportverein) viel diverser sind als jene die die Grundversicherung (meist ausgestellt durch Ärzte und Spitäler nach TARMED Standard) betreffen, liegt die Vermutung nahe, dass im Bereich der Zusatzversicherung ein hoher Anteil der Verwaltungskosten der Rechnungsprüfung zugeschrieben werden kann.

Die AXA profitiert bei einer automatisierten Indexierung von Rechnungen nicht nur von einer Kostensenkung, sondern kann auch einen Vorteil für ihre Kundinnen und Kunden generieren. Die Kundinnen und Kunden erhalten durch die automatisierte Verarbeitung der eingereichten Rechnungen schneller das geforderte Geld.

### 4.1.1 Aktueller Prozess der Rechnungseinreichung

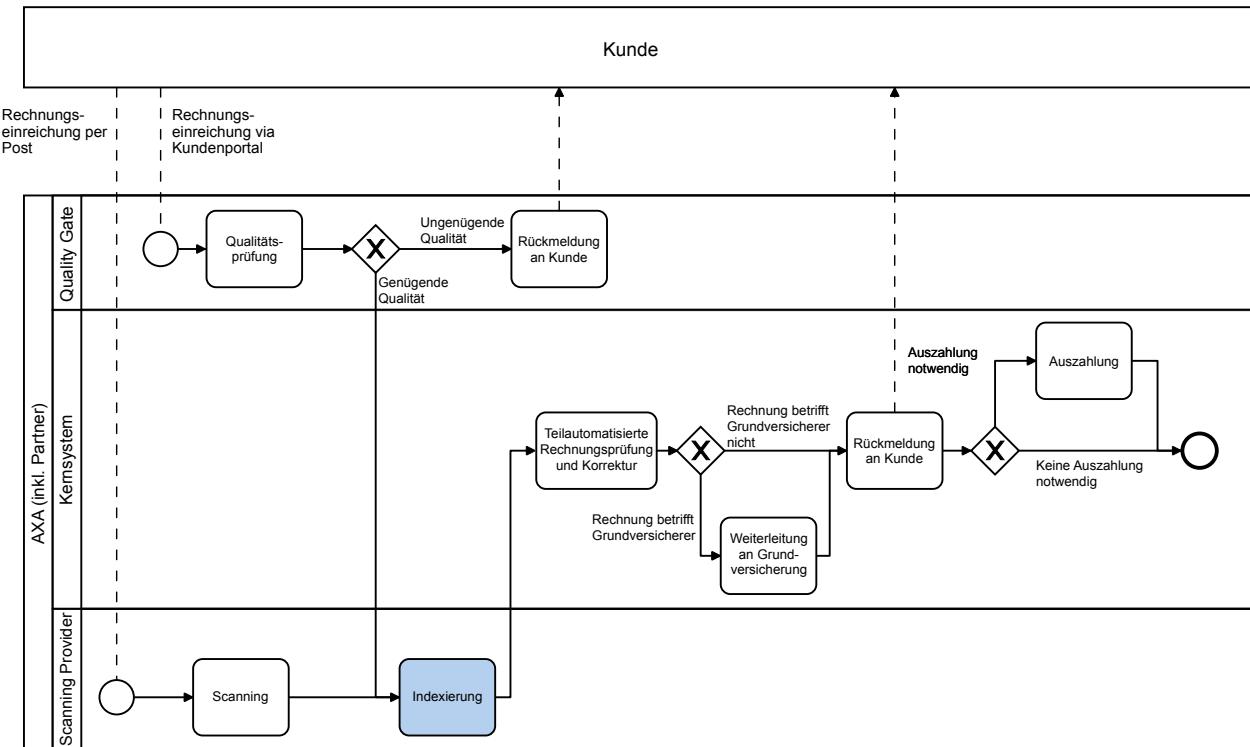
Der Prozess der Rechnungseinreichung und -verarbeitung (vgl. Abbildung 27) der AXA kann aufgrund der zwei im vorherigen Kapitel erwähnten Ereignisse angestossen werden. Die Kundin oder der Kunde kann entweder digital oder per Post eine Rechnung einreichen.

Im Kundenportal hat die Kundin oder der Kunde die Möglichkeit eine Rechnung hochzuladen. Dabei kann entweder ein Dokument auf dem Endgerät gewählt oder die Kamera des Gerätes genutzt werden, um eine Rechnung zu fotografieren.

## 4.1 Einführung in das Fallbeispiel

---

Abbildung 27: Prozess der Rechnungseinreichung und -verarbeitung der AXA Gesundheitsvorsorge



Nach erfolgreichem Hochladen der Rechnung im Kundenportal durchläuft diese eine erste manuelle Qualitätsprüfung. Diese Qualitätsprüfung wurde eingeführt, da hochgeladene Rechnung teilweise ungenügende Qualität vorwiesen. Entspricht die Rechnung nicht den Qualitätsanforderungen oder fehlt eine Seite oder eine ärztliche Verordnung, so wird die Kundin oder der Kunde gebeten, die vollständige Rechnung erneut hochzuladen.

Nach erfolgreicher Qualitätsprüfung wird die Rechnung an den Scanning und Indexierungsdienstleister der AXA weitergeleitet.

Entscheidet sich die Kundin oder der Kunde für die Einreichung per Post, so wird der Brief direkt an den Scanning und Indexierungsdienstleister der AXA weitergeleitet. Dieser Dienstleister scannt die weitergeleitete Rechnung ein.

Nach beiden dieser Einstiegspunkten in den Prozess indexiert der Scanning und Indexierungsdienstleister die eingereichte Rechnung. Dieser Aufgabenschritt erfolgt teilweise automatisiert und teilweise manuell. Wie genau die Indexierung abläuft ist nicht bekannt. Der genaue Ablauf dieser eingekauften Dienstleistung wird als Geschäftsgeheimnis gewahrt.

Nach der Indexierung werden die Scans sowie das strukturierte Resultat der Indexierung elektronisch an das Kernsystem der AXA Gesundheitsvorsorge übermittelt. Dieses Kernsystem verarbeitet die eingegangenen Rechnungen aufgrund eines Regelwerks. Kann die Rechnung nicht verarbeitet werden, weil diese nicht korrekt indexiert wurde oder Informationen fehlen, muss eine Fachspezialistin oder ein Fachspezialist eingreifen. Nach allfälligen Rückfragen und Korrekturen wird die Rechnung verarbeitet.

Nach der erfolgreichen Verarbeitung der Rechnung wird die Kundin oder der Kunde elektronisch informiert, ob die beanspruchten Leistungen versichert sind und ob eine Rückvergütung ausbezahlt wird.

Hat die Kundin oder der Kunde die AXA bevollmächtigt, so wird die Rechnung, je nach Rechnungspositionen, automatisiert an die jeweilige Grundversicherung weitergeleitet.

Ziel der AXA Gesundheitsvorsorge ist es, diesen Prozess für eingereichte Rechnungen, welche Fitnesscenter und Optiker betreffen, vollständig zu automatisieren. Dabei gibt es in diversen Bereichen Herausforderungen, welche aktuell angegangen werden.

Ausgangslage für den Arbeitsschritt der Indexierung ist eine digitalisierte Rechnung. Ziel der Indexierung ist es, aus dieser Rechnung strukturierte Informationen zu gewinnen und an das Kernsystem der AXA zu übermitteln. In dieser Arbeit wird die Extraktion dieser strukturierten Daten aus den digitalisierten Rechnungen mit Hilfe künstlicher Intelligenz behandelt. Das Übermitteln der strukturierten Informationen an das Kernsystem ist bereits gelöst und aus diesem Grund nicht Teil dieser Arbeit.

### 4.2 Anforderungen

Um Rechnungen von Fitnesscentern und Optikern automatisiert verarbeiten zu können, müssen diese Rechnungen als solche klassifiziert und die notwendigen Informationen, welche nachfolgend genauer spezifiziert werden, extrahiert werden.

## 4.2 Anforderungen

Um eine Rechnung eines Optikers zu verarbeiten, sind folgende Informationen relevant:

- **Leistungsbezüger**

Es muss ermittelt werden, für wen die Rechnung ausgestellt wurde. Anhand dieser Information wird geprüft, ob und wie diese Person bei der AXA versichert ist. Auch wird damit geprüft, dass der maximal versicherte Betrag noch nicht ausgeschöpft ist.

Ist der Leistungsbezüger minderjährig, so wird ein gewisser Betrag von der Grundversicherung übernommen. In diesem Fall wird dieser Betrag von der Rückvergütung der AXA abgezogen und die Rechnung an die Grundversicherung weitergeleitet.

- **Totalbetrag der Rechnung (inkl. Währung)**

Dieser Betrag bildet die Grundlage zur Berechnung der geschuldeten Leistung an die Kundin oder den Kunden.

Einzelne Rechnungspositionen sind für Rechnungen von Optikern nicht relevant.

- **Hinweis auf eine ärztliche Verordnung**

Besteht eine ärztliche Verordnung, ist ein gewisser Betrag über die Grundversicherung versichert. In diesem Fall wird dieser Betrag von der Rückvergütung der AXA abgezogen und die Rechnung an die Grundversicherung weitergeleitet.

Folgende Informationen sind notwendig, um eine Rechnung für ein Fitness-Abonnement zu verarbeiten:

- **Leistungsbezüger**

Es muss ermittelt werden, für wen die Rechnung ausgestellt wurde. Anhand dieser Information wird geprüft ob und wie diese Person bei der AXA versichert ist. Auch wird damit geprüft, dass der maximal versicherte Betrag noch nicht ausgeschöpft ist.

- **Totalbetrag der Rechnung (inkl. Währung)**

Dieser Betrag bildet die Grundlage zur Berechnung der geschuldeten Leistung an die Kundin oder den Kunden.

Einzelne Rechnungspositionen sind für Rechnungen von Fitnesscentern nicht relevant.

- **Fitnesscenter (Leistungserbringer)**

Die AXA anerkennt alle Fitnesscenter mit dem Label Qualitop von Qualicert oder mit einer Bewertung von mindestens 3 Sternen bei Fitnessguide.

Anhand dieser Informationen können Rechnungen automatisiert verarbeitet werden. Es muss sichergestellt werden, dass alle relevanten Informationen korrekt extrahiert werden, denn durch die Automatisierung entfällt jegliche manuelle Prüfung. Fehler würden, wenn überhaupt, erst der Kundin oder dem Kunden auffallen.

### 4.3 Vorgehen und Methodik

Wie in den Anforderungen an die Indexierung bereits angemerkt, müssen zwei Teilschritte, die Klassifizierung und die Extraktion von Informationen, analysiert werden. Dabei kommt ein explorativer Ansatz zur Anwendung. Für beide Teilschritte werden separate Experimente durchgeführt. Der Ablauf der Experimente ist identisch.

Für die Experimente werden zuerst Testdaten aus dem System der AXA extrahiert und aufbereitet. Es wird in jedem Experiment mindestens eine künstliche Intelligenz geschaffen. Die Resultate der Experimente werden jeweils unabhängig voneinander diskutiert. Es wird das Potential der gewählten Ansätze sowie mögliche weitere Ansätze diskutiert.

Aus der Diskussion der Experimente werden Schlussfolgerungen für die Anwendbarkeit der künstlichen Intelligenz zur Indexierung von Rechnungen abgeleitet.

Das explorative Vorgehen wurde gewählt, da unzählige Testdaten zur Verfügung stehen. In diesem Fall kann mit einem explorativen Ansatz schnell evaluiert werden, ob ein bestimmter Ansatz erfolgversprechend ist oder nicht.

Ein weiterer Grund für die Wahl des explorativen Vorgehens ist, dass sich der Autor während der Erstellung dieser Arbeit, im speziellen während der Durchführung der Experimente, das Wissen im Bereich der künstlichen Intelligenz erst aneignen musste.

### 4.4 Teil 1 - Klassifizierung von Rechnungen

In diesem Kapitel werden zwei Experimente erläutert, mit welchen die Klassifizierung von Rechnungen ermöglicht werden soll. Die Klassifizierung dient dazu, eine Rechnung einer bestimmten Art (Klasse) zuzuweisen. Die Klassen Optiker und Fitness sind durch die Anforderungen gegeben. Neben diesen Klassen könnten Rechnungen noch in weitere Klassen eingeteilt werden. Um die Erweiterbarkeit des Modells um weitere Klassen zu prüfen, wird die zusätzliche Klasse Sportverein eingeführt. Außerdem wird die Klasse Andere eingeführt, damit das Modell Rechnung einer unbekannten Klasse aussteuern kann. In diesem Experiment werden Rechnungen in die Klassen Optiker, Fitness, Sportverein und Andere eingeteilt. Zu einem späteren Zeitpunkt wäre es denkbar, die Anzahl Klassen zu erhöhen und somit auch andere Arten von Rechnungen zu automatisieren.

Der Ausgangspunkt für die Klassifizierung einer Rechnung ist ein Bild. Im Kapitel 4.4.1 wird ein Ansatz zur Klassifizierung von Rechnungen erläutert, welcher auf existierenden Modellen zur Klassifizierung von Bildern basiert.

Im Kapitel 4.4.2 wird ein Text-basierter Ansatz zur Klassifizierung der Rechnungen erläutert.

Die beiden Ansätze zur Klassifizierung werden anhand der knapp 24'500 bereits bei der AXA eingereichten Rechnungen evaluiert. Um dies zu ermöglichen, wurden die Rechnungen aus dem System der AXA exportiert und in die oben genannten Klassen eingeteilt.

Der Datensatz musste auf 17'196 Rechnungen reduziert werden. Etwas mehr als 7'000 Rechnungen hatten mehr als eine Seite. Aus diesem Grund konnten diese Rechnungen mit den vorhandenen Informationen nicht eindeutig einer Klasse zugewiesen werden. Eine manuelle Klassifizierung würde für den Umfang dieses Experiments einen zu grossen Arbeitsaufwand darstellen.

Bei der Durchsicht der Rechnungen wurden 127 Rechnungen aufgrund mangelnder Qualität aussortiert. Diese Problematik wird durch die vor kurzem eingeführte Qualitätsprüfung nicht mehr vorkommen und ist deshalb für diese Arbeit nicht weiter relevant.

Nach der Einteilung in die vier Klassen ist festzustellen, dass die 17'196 Rechnungen eine sehr unregelmässige Verteilung aufweisen (vgl. Abbildung. 28). Die Klasse Sportverein ist gegenüber den anderen Klassen mit nur 760 Rechnungen unterrepräsentiert. Die Klasse Andere ist hingegen mit 12'751 Rechnungen überrepräsentiert. Es ist wichtig, diesen Umstand während dem Training eines Modells zu berücksichtigen. Ansonsten würde die Voraussage des Modells zu oft in der überrepräsentierten Klasse Andere resultieren (Buda, Maki & Mazurowski, 2017). Diese Problematik kann auf diverse Arten angegangen werden. In dieser Arbeit wird dazu eine, aufgrund der Klassenverteilung, gewichtete Loss Funktion verwendet.

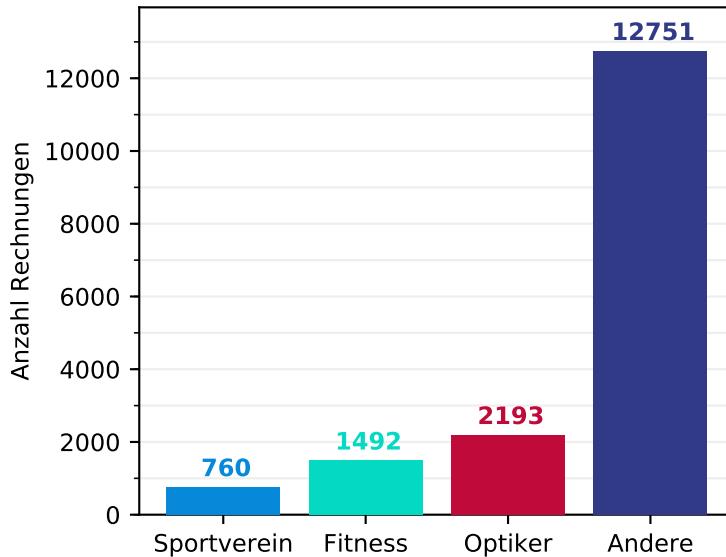
### 4.4.1 Bild-basierte Rechnungsklassifizierung

Dieses Kapitel erläutert ein Experiment, bei welchem Algorithmen und Modelle aus dem Bereich der Computer Vision angewendet werden, um Rechnungen zu klassifizieren.

Im Folgenden wird das ResNet, das Inception-ResNet-V2 sowie das NASNet large Netzwerk angewendet, um die bisher bei der AXA eingereichten Rechnung zu klassifizieren.

Um Objekterkennungsmodelle mit Millionen von Parametern zu trainieren, werden viele Trainingsdaten und eine enorme Kapazität an Rechenleistung benötigt. Das Konzept des Transfer Learning bietet die Möglichkeit, diese beiden Problematiken zu Umgehen und dabei nur wenige oder keine Einbussen bei der Trefferquote verzeichnen zu müssen (vgl. Kapitel 3.14).

Abbildung 28: Ungleichverteilung der Klassen innerhalb des Trainingsdatensatzes



Um die Rechnungen zu klassifizieren wird Transfer Learning angewendet, indem die genannten Modelle auf dem ImageNet Datensatz trainiert werden, bevor die eigentliche Problemstellung angegangen wird.

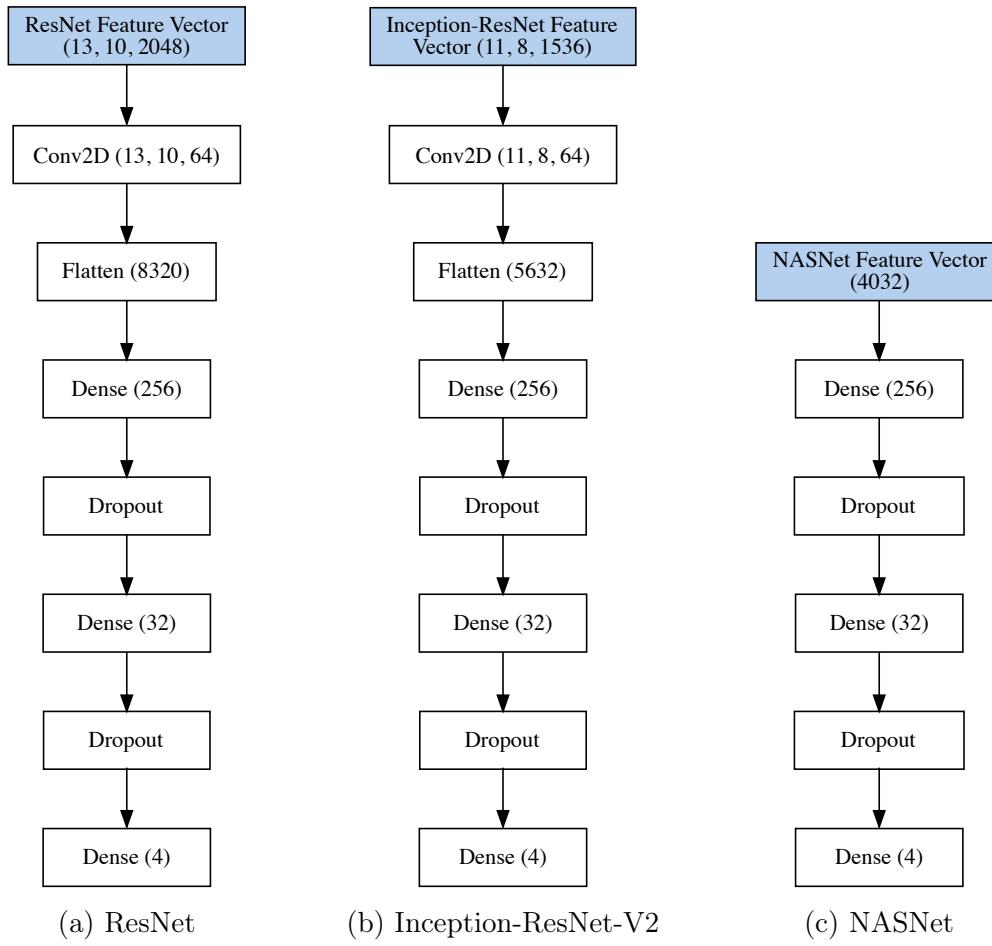
Nach dem Training auf dem ImageNet Datensatz werden die letzten Schichten des Netzwerks, jene die für die Klassifizierung zuständig sind, durch ein neues Klassifizierungsnetzwerk ersetzt. Dadurch wird der Trainingseffekt durch den ImageNet Datensatz beibehalten und die Klassifizierung so angepasst, dass sie die Einteilung in die vier vorliegenden Klassen erlaubt.

Die Abbildung 29 zeigt die Klassifizierungsmodelle, welche zur Anwendung kommen. Input ist jeweils der Feature-Vektor, welcher durch das ResNet, Inception-ResNet beziehungsweise NASNet Netzwerk erstellt wurde. Beim ResNet und Inception-ResNet wird dieser Feature-Vektor durch ein Convolution und ein Flatten Layer in einen eindimensionalen Vektor gebracht. Das NASNet liefert bereits einen eindimensionalen Feature-Vektor und die ersten zwei Schichten entfallen. Durch zwei Hidden Layer wird der Vektor schlussendlich zu einem One-Hot Encoded Vektor für die vier Klassen reduziert. Zwischen den Hidden Layern sind zwei Dropout Layer zur Reduzierung des Overfitting eingeschoben.

Die genannten Modelle werden mit 80% der vorhandenen Daten trainiert, die übrigen 20% werden benötigt, um den Trainingsfortschritt zu prüfen. Mit diesen 20% Testdaten soll ein allfälliges Overfitting erkannt werden.

Die Abbildung 30 zeigt das Training der genannten Modelle während 60 Trainingsepochen (Trainungseinheiten). Die Abbildungen 30a und 30b zeigen die Trefferquote respektive das Loss während dem Training. Die Abbildungen 30c und 30d zeigen die Trefferquote respektive das Loss bei der Anwendung des Modells auf den Testdaten.

Abbildung 29: neuronale Netze, welche bei der Bild-basierten Klassifizierung zur Anwendung kommen

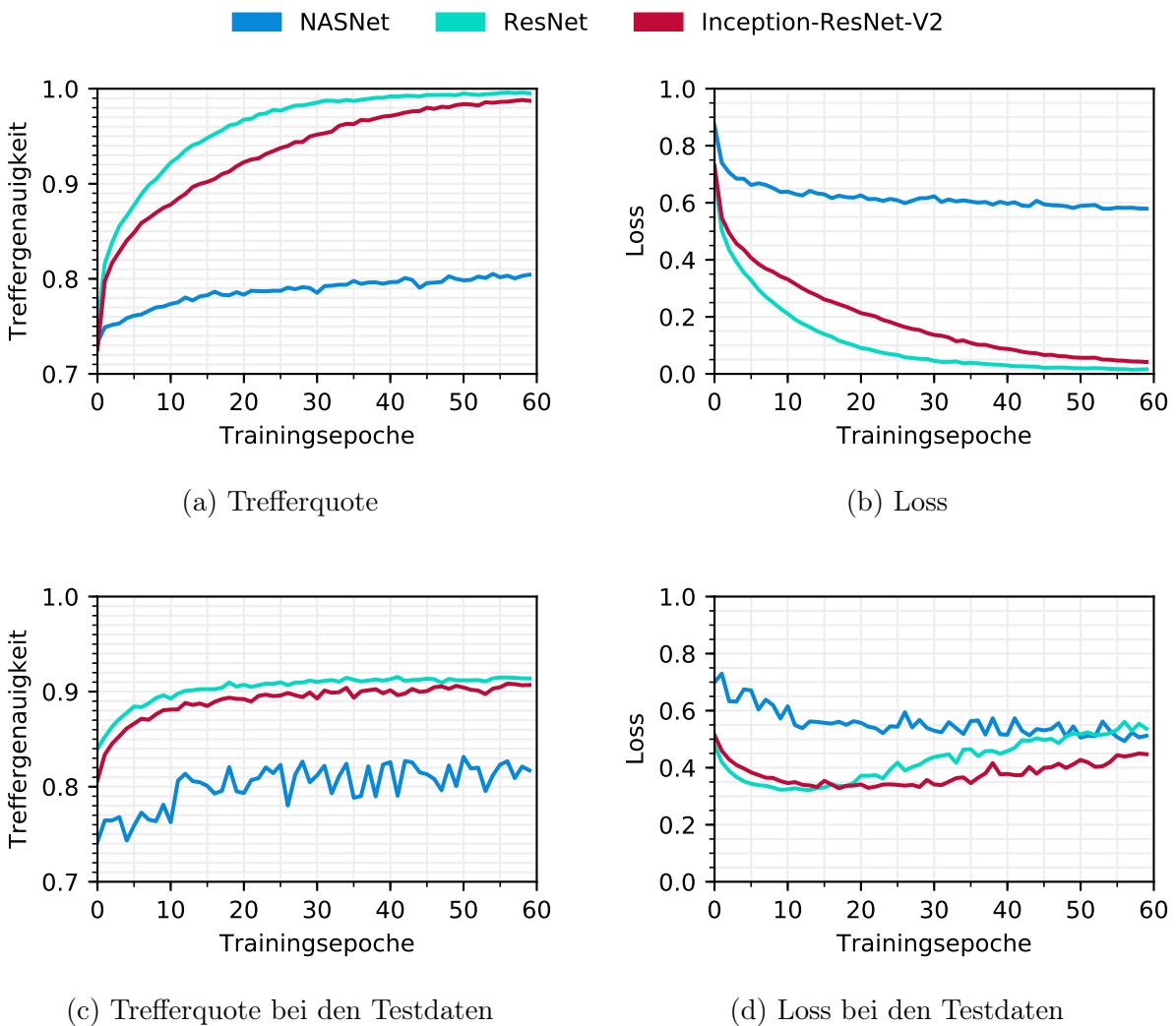


Auffällig ist das schlechte Resultat des NASNet Modells. Obwohl das NASNet als eines der genauesten Klassifizierungsmodelle für Bilder gilt, schneidet es im vorliegenden Experiment mit Abstand am schlechtesten ab. Wie in Abbildung 30c ersichtlich ist, liegt die Trefferquote ungefähr bei 0.8 und somit knapp 0.2 unter den beiden anderen Modellen.

Eine fundierte Erklärung, weshalb das NASNet Modell so schlecht abschneidet, kann nicht gefunden werden. Eine mögliche Erklärung ist der mit Abstand kleinere Feature-Vektor des NASNets. Dieser ist mit nur 4'032 Neuronen wesentlich kleiner als jener des ResNet (26'6240 Neuronen) und des Inception-ResNet-V2 (13'5168 Neuronen).

Die Ergebnisse des ResNet und des Inception-ResNet-V2 sehen dagegen wesentlich besser aus. Die Modelle weisen bereits nach 30 Epochen ein Bias von weniger als 5% (ResNet) respektive weniger als 2% (Inception-ResNet-V2) auf. Nach 60 Epochen Training weisen die beiden Modelle ein Bias von weniger als 1.5% respektive 0.5% auf. Dies zeigt, dass die gewählten Modelle lernen und grundsätzlich geeignet sind, die Problemstellung anzugehen.

Abbildung 30: Statistiken aus dem Training der Bild-basierten Klassifizierung von Rechnungen



Neben der auch nach 60 Epochen noch leicht steigenden Trefferquote hat das Loss auf den Testdaten (vgl. Abbildung 30d) bereits nach 25 (ResNet) respektive 15 (Inception-ResNet-V2) Epochen den Wendepunkt erreicht. Dies zeigt, dass das Modell zu wenig gut generalisiert. Die steigende Testgenauigkeit vermittelt zwar einen guten Eindruck, das steigende Loss zeigt jedoch, dass das Modell in seinen Entscheidungen immer unsicherer wird, das heisst die Wahrscheinlichkeit, mit welcher sich das Modell sicher ist, eine Rechnung einer bestimmten Klasse zuzuweisen, sinkt. Das Modell beginnt also nach nur wenigen Epochen auswendig zu lernen (Overfitting).

Das Problem des Overfitting könnte womöglich durch zusätzliche Trainingsdaten gemindert werden, diese stehen jedoch nicht zur Verfügung.

#### 4.4.2 Text-basierte Rechnungsklassifizierung

Werden die Rechnungen nicht als Bilder angesehen, sondern wird der auf Ihnen aufgedruckte Text als zentraler Aspekt angesehen, so liegt die Klassifizierung der Rechnung aufgrund dieses Textes nahe. In diesem Kapitel wird ein Experiment mit einem Text-basierten Ansatz zur Klassifizierung von Rechnungen erläutert.

Um die Rechnungen aufgrund des enthaltenen Textes zu klassifizieren, muss dieser aus den Bildern der Rechnungen extrahiert werden. Dies wird mit Hilfe von Tesseract OCR, einem OCR System, welches selbst auf einem hoch komplexen neuronalen Netzwerk basiert, gemacht. Tesseract OCR wurde gewählt, da es frei zugänglich und dem Autoren bekannt ist.

Nach der Extraktion der Texte aus den Rechnungen wird aus diesen ein Wörterbuch gebildet. Dieses Wörterbuch wird als Methode zum Word embedding verwendet. Das Word embedding dieses Wörterbuches ist sehr einfach gehalten und erlaubt keine Rückschlüsse auf die Bedeutung der Wörter aufgrund des resultierenden Vektors.

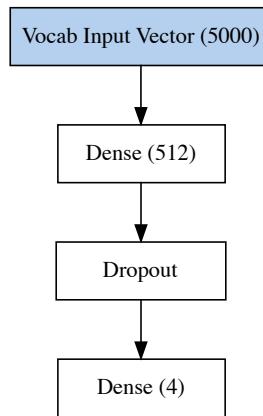
Das Wörterbuch wird erstellt, indem der Text erst in Kleinbuchstaben umgewandelt und anschliessend bei Leerzeichen getrennt wird. Es werden alle Stopp-Wörter wie *und*, *ein* und *diese* entfernt, da aus ihnen keine Informationen gewonnen werden können. Es werden die häufigsten Wörter ermittelt, welche später für das Word embedding verwendet werden. Das Wörterbuch behält sich nur die häufigsten Wörter, um die Komplexität gering zu halten. Die optimale Anzahl Wörter wird zu einem späteren Zeitpunkt ermittelt.

Nach dem Wörterbuch wird ein Klassifizierungsmodell erstellt. Dieses Modell wird bewusst sehr einfach gehalten und könnte bei Bedarf, beispielsweise bei einem hohen Bias, erweitert werden. Input dieses Netzwerks ist ein Vektor in der Länge der Anzahl Wörter im Wörterbuch. Pro Wort wird in diesem Vektor die Präsenz beziehungsweise Absenz des Wortes innerhalb einer Rechnung angegeben. Nach dem Input folgt ein Fully Connected Layer als Hidden Layer. Dieser Hidden Layer ist durch einen Dropout Layer mit dem Fully Connected Output Layer verbunden. Der Output Layer klassifiziert die Rechnung mit Hilfe eines one-hot encoded Vektors (vgl. Abbildung 31).

Während dem Experiment wurde die optimale Grösse des Wörterbuchs durch Ausprobieren ermittelt. Das Bias ist bei den meisten der verwendeten Grössen des Wörterbuchs sehr gering. Der gewählte Ansatz scheint die Problemstellung bereits mit einem kleinen Vokabular gut bewältigen zu können.

Bei der Exploration der Grösse des Wörterbuchs wurde festgestellt, dass das Modell zur Klassifizierung mit einem grösseren Wörterbuch von 5'000 Wörtern die besten Ergebnisse, sprich die kleinste Varianz, liefert. Je grösser das Wörterbuch, desto schneller hat das Modell begonnen auswendig zu lernen (Overfitting). Um diesem Effekt entgegenzuwirken wurde ein grosser Dropout von 0.92 gewählt. Auch dieser Wert wurde explorativ ermittelt.

Abbildung 31: Neuronales Netzwerk, welches bei der Text-basierten Klassifizierung zur Anwendung kommt



Bei dem gewählten Modell, mit einem Vokabular von 5'000 Wörtern und einem Dropout von 0.92, erreicht das Loss auf den Testdaten nach 23 Epochen Training den Wendepunkt (vgl. Abbildung 32d). Dabei wird eine Trefferquote von 98.4% erzielt (vgl. Abbildung 32c). Zu diesem Zeitpunkt beginnt das Modell auswendig zu lernen. Es ist also nicht sinnvoll das Modell länger auf den vorliegenden Daten zu trainieren.

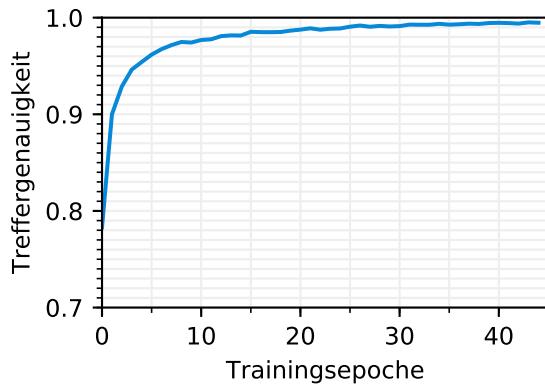
Bei diesen Ergebnissen liegt es nahe, dass der Ansatz durch etwas Optimierung noch weiter verbessert werden kann. Um dieses Optimierungspotential genauer zu definieren, wird im Folgenden eine Fehleranalyse durchgeführt. Es wird aufgezeigt, worauf bei der Klassifizierung im vorliegenden Fallbeispiel besonders geachtet werden muss. Anschliessend werden aus den falsch klassifizierten Rechnungen Verbesserungsvorschläge abgeleitet.

Die falsche Klassifizierung einer Rechnung ist dann problematisch, wenn diese einer Klasse zugewiesen wird, welche automatisch verarbeitet wird. Die Confusion Matrix in Abbildung 33 zeigt, dass das Text-basierte Modell sechs aus 3'367 Rechnungen des Testsets fälschlicherweise als Rechnungen eines Fitnesscenters und fünf Rechnungen fälschlicherweise als Rechnungen für einen Sportverein klassifiziert hat. Diese Rechnungen würden automatisiert als solche abgerechnet werden, was der Kundin oder dem Kunden nicht vorenthalten bleibt. Die Kundin oder der Kunde müsste in diesem Fall aktiv werden, damit der Fehler bemerkt wird.

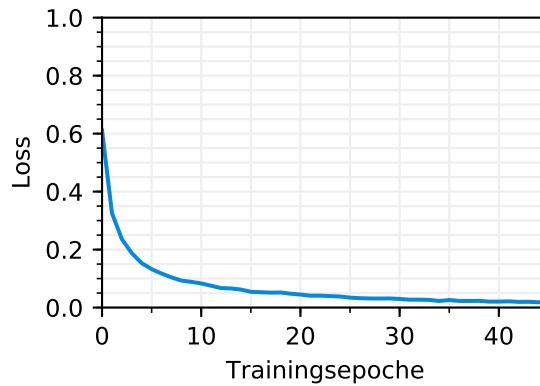
Würde eine Rechnung fälschlicherweise als Andere klassifiziert werden, so muss eine manuelle Verarbeitung der Rechnung stattfinden. In diesem Fall merkt die Kundin oder der Kunde nichts von dem Fehler.

Aus den genannten Gründen ist die Genauigkeit für das vorliegende Beispiel die wichtigste Metrik. Die ermittelte Genauigkeit liegt bei 97.86% (Fitness), 100% (Optiker) respektive 96.50% (Sportverein).

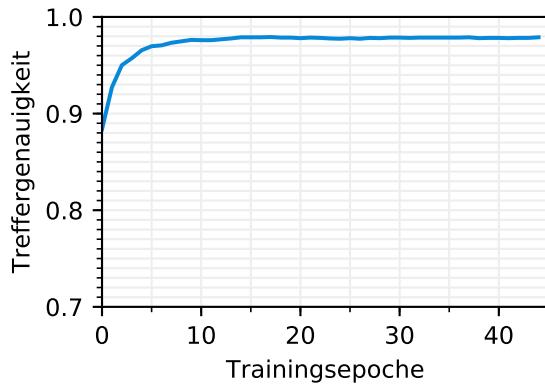
Abbildung 32: Statistiken aus dem Training der Text-basierten Klassifizierung von Rechnungen



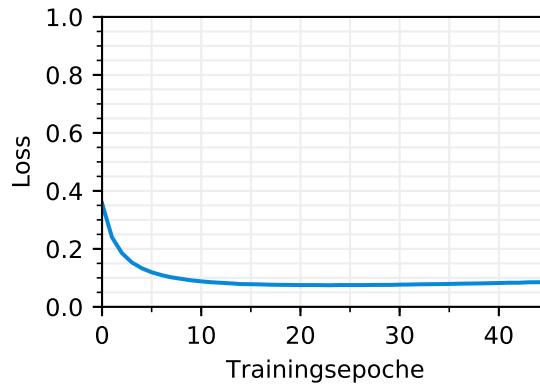
(a) Trefferquote



(b) Loss



(c) Trefferquote bei den Testdaten

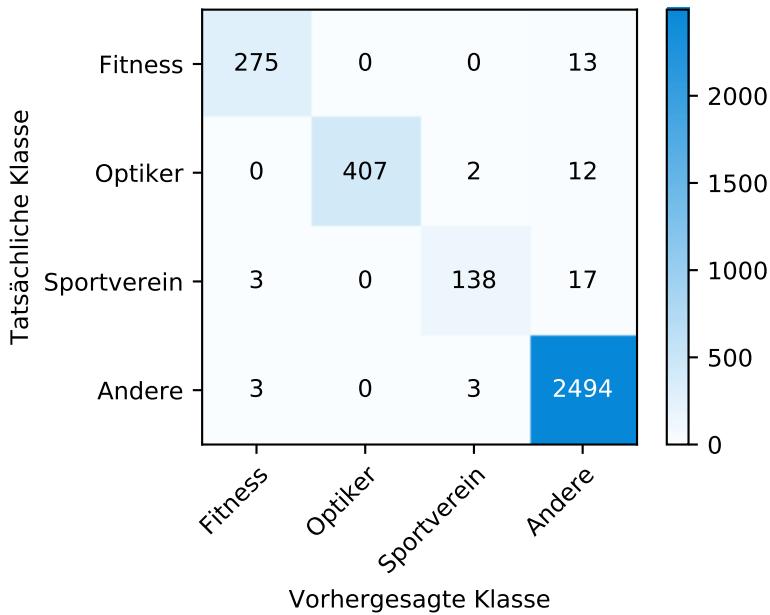


(d) Loss bei den Testdaten

Die insgesamt 51 falsch klassifizierten Rechnungen aus dem Testdatensatz sowie die während dem Training falsch klassifizierten Rechnungen wurden einer Fehleranalyse unterzogen. Ng (2018) suggerieren für die Verbesserung eines Modells dort anzusetzen, wo am meisten falsche Klassifikationen vorliegen. Dabei wird argumentiert, dass dort das Potential, die Trefferquote zu erhöhen, am höchsten ist. In unserem Fall ist die Trefferquote aber eher nebenschlüssig. Die 42 aus dem Testset fälschlicherweise als Andere klassifizierten Rechnungen haben keinen grossen negativen Einfluss auf den Geschäftsprozess. Aus diesem Grund fokussiert sich die Analyse auf die fälschlicherweise als eine der automatisierbaren Klassen klassifizierten Rechnungen.

Während der Analyse wurden folgende Probleme und Verbesserungsmöglichkeiten identifiziert:

Abbildung 33: Confusion Matrix nach 23 Trainingsepochen des Text-basierten Modells zur Klassifizierung von Rechnungen



**Ungenauigkeiten des Optical Character Recognition:** Ein Problem, welches sofort ins Auge sticht, sind die Ergebnisse des Optical Character Recognition Systems. Durch die mässige Qualität fotografiert Rechnungen können gewisse Wörter nur schlecht oder gar nicht erkannt werden.

Durch diese Ungenauigkeiten während dem OCR Prozess entstehen viele Wörter mit Schreibfehlern. Das Wörterbuch erkennt aber nur Wörter, welche mit den gelernten Wörtern identisch sind. Wurde während dem Training des Wörterbuchs ein Wort nie in einer gewissen falschen Schreibweise angetroffen, so ist dieses Wort für das Klassifizierungsmodell nutzlos.

Eine Problematik für das Optical Character Recognition System ist eine schlechte Ausleuchtung des Fotos einer Rechnung. In diesem Fall erkennt das OCR System nur sehr wenig bis überhaupt keinen Text. Dadurch kann die Rechnung nicht klassifiziert werden.

Eine weitere Problematik ist ein körniges Bild. In diesem Fall erkennt das OCR System diakritische Zeichen wie ein é, wo keine sind. Dadurch kann ein Wort nicht im Wörterbuch gefunden werden und es trägt somit nicht zur Klassifizierung bei.

Um die genannten Probleme anzugehen und die Qualität der Resultate aus dem OCR System zu verbessern, stehen mehrere Optionen zur Verfügung. Diese werden im Folgenden erläutert.

Das Modell des OCR Systems, welches selbst auch ein neuronales Netzwerk ist, sollte auf den Rechnungen trainiert werden. Auch hier kommt das Konzept des Transfer Learning zum Einsatz. Das verwendete OCR System (Tesseract) erlaubt es nicht nur ganz eigene Modelle zu trainieren, sondern ermöglicht auch das Fine-tuning der mitgelieferten Modelle. In vorliegenden Fall würde ein Fine-tuning der mitgelieferten Modelle von Tesseract die Qualität der Ergebnisse womöglich stark verbessern, wie dies ein Beispiel im Tesseract Wiki suggeriert. Im Wiki wird erwähnt, dass das OCR Modell auf bisher unbekannten Schriftarten trainiert werden sollte. Auch könnten durch das Fine-tuning einige Zeichen, welche Tesseract aktuell unbekannt sind, erlernt werden (O.V., 2016).

Im Wiki von Tesseract wird neben dem Fine-Tuning des OCR Modells auch die Aufbereitung der Bilder als zentraler Aspekt der Verbesserung der Qualität der OCR Resultate genannt. Unter der Aufbereitung versteht das Tesseract Wiki das Vergrössern, die Binärisierung, das Rotieren und Entzerren der Bilder sowie das Entfernen von Rauschen und dunkeln Rändern durch das Scanning (O.V., 2019).

In diesem Experiment ist die Grösse der vorliegenden Bilder kein Problem. Die meisten Bilder wurden in einer grossen Auflösung aufgenommen. Würde hier ein Problem vorliegen, so müsste beim digitalen Einreichen der Rechnungen über das Kundenportal, eine minimale Auflösung forciert werden.

Die Binärisierung, sprich das Umwandeln in Schwarz-Weiss-Bilder, der Rechnungen scheint durchaus ein Problem darzustellen. Wie bereits erwähnt, sind die OCR Resultate dann besonders schlecht, wenn das Bild schlecht ausgeleuchtet ist. In diesem Fall ist der Binärisierungs-Algorithmus, welcher Tesseract standardmässig anwendet, nicht geeignet. Mit einem geeigneteren Algorithmus zur Binärisierung könnte die Qualität der OCR Ergebnisse verbessert werden. Die Bilder, welche in einem schlechten OCR Ergebnis resultieren, könnten durch verschiedene Binärisierungs-Algorithmen umgewandelt werden. Es kann dann jenes dieser umgewandelten Bilder verwendet werden, welches die besten OCR Resultate liefert.

Rotierte oder verzerrte Bilder sind im vorliegenden Experiment kein Problem. Die Rechnungen sind bereits richtig rotiert und kaum verzerrt.

Ein körniges Bild ist, wie erwähnt, aktuell ein Problem, welches die Qualität des OCR Ergebnis stark beeinflusst. Das Entfernen dieses Rauschens (Körnung) im Bild würde die Qualität des OCR Ergebnis und somit die Genauigkeit des Klassifizierungsmodells verbessern.

Scanning Ränder, wie diese auf dem Tesseract Wiki angemerkt werden, sind im vorliegenden Fall kein Problem. Ein ähnliches Problem, welches sich aber äussert, sind Ränder auf Fotografien von Rechnungen. Fotografiert jemand eine Rechnung, so ist meist noch etwas vom Hintergrund, beispielsweise einem Holztisch, zu sehen. Diese Ränder stören die Qualität der OCR Ergebnisse indirekt, durch eine verschlechterte Binärisierung, sowie direkt durch erkannte Buchstaben, wo eigentlich nur ein Hintergrund zu sehen ist. Es ist also zentral, diese Ränder zu entfernen. Viele Apps von Krankenversicherern bieten bereits eine solche Funktion an, durch welche Ränder automatisch erkannt werden und der Nutzer die Möglichkeit hat, diese anzupassen. Eine solche Funktion ist für das vorliegende Fallbeispiel empfehlenswert.

Mit den genannten Ansätzen können die Resultate des OCR Systems verbessert werden. Es kann jedoch nicht davon ausgegangen werden, dass das OCR System immer alles korrekt erkennt. Aus diesem Grund ist eine Nachbearbeitung des OCR Resultates sinnvoll. Im Kapitel 3.9 wurde ein Modell, basierend auf LSTM Netzwerken vorgestellt, welches darauf trainiert wird, Rechtschreibfehler zu korrigieren. Dieses Modell dürfte auch auf die Korrektur von OCR Fehlern anwendbar sein. Durch die Korrektur der OCR Fehler vor der Anwendung des Klassifizierungsmodells kann die Qualität dieses Modells erhöht werden. Mit einem Domain-Know-How basierten Ansatz zur Korrektur von Fehlern erreichen Sorio, Bartoli, Davanzo und Medvet (2012) eine knappe Verdoppelung der Qualität der Resultate aus dem OCR System. Selbst für qualitativ sehr schlechte Bilder werden sehr gute Resultate erreicht.

**Überrepräsentation einzelner Wörter in gewissen Klassen:** Die Fielmann AG ist wohl einer der bekanntesten Optiker in der Schweiz. Bei einer Rechnung von Fielmann denken wir automatisch an eine Rechnung eines Optikers. Genau so scheint auch unser Modell zu denken, denn eine Rechnung von Fielmann wird stets als Rechnung von einem Optiker klassifiziert. Dies stellt aber eine Herausforderung dar, denn Fielmann verkauft nicht nur Seh- sondern auch Hörlhilfen.

Diese Problematik erinnert an das Problem der überrepräsentierten Klassen. Auf Stufe der Klassen trägt die Gewichtung der Loss-Funktion der Überrepräsentation einzelner Klassen Rechnung. Auf der Stufe des Vokabulars wird diesem Ungleichgewicht bisher keine Rechnung getragen.

Um diese Problematik zu lösen sind zwei Ansätze denkbar. Zum einen könnte das Wort Fielmann aus dem Wörterbuch ausgeschlossen werden. Dies könnte aber einen sehr negativen Einfluss auf die Klassifizierung tatsächlicher Rechnungen von Optikern haben. Zum anderen ist ein Over- oder Undersampling, also das künstliche vermehren oder vermindern, der unter- beziehungsweise überrepräsentierten Rechnungen denkbar. Das bedeutet, jene Fielmann Rechnungen welche Hörlhilfen betreffen, sollten dem Modell während dem Training öfters vorgelegt werden. Damit würde das Modell auf diese Art von Rechnungen sensibilisiert werden.

**Nicht eindeutige Wahrscheinlichkeiten bei der Klassifizierung:** Die verwendete Methode der Klassifizierung verwendet vier Wahrscheinlichkeiten, mit welcher eine Rechnung einer bestimmten Klasse angehört. In Summe ergeben diese Wahrscheinlichkeiten immer 1. Die Klasse, bei welcher die Wahrscheinlichkeit am höchsten liegt, gewinnt. Es kann vorkommen, dass die Wahrscheinlichkeiten annähernd gleichmäßig verteilt sind und eine Rechnung mit einer Klassenwahrscheinlichkeit von nur 25.1% klassifiziert wird.

Um diese Problematik zu lösen, sollte eine minimale Wahrscheinlichkeit definiert werden, mit welcher eine Rechnung klassifiziert werden muss. Wahrscheinlichkeiten, welche unter diesem Minimum liegen, dürfen nicht verwendet werden. Dieser Ansatz hat neben der Reduzierung der Fehlerquote aber auch eine Reduzierung der Automatisierung zur Folge. Es muss abgewogen werden, ob die AXA Gesundheitsvorsorge das Risiko einer höheren Fehlerquote eingehen möchte und dafür mehr Rechnungen automatisieren kann oder nicht.

**Klassifizierung aufgrund irrelevanter Wörter:** Bei der Betrachtung einiger Rechnungen wurde festgestellt, dass diese hauptsächlich aus einem Einzahlungsschein bestehen. Dies kam besonders oft bei Sportvereinen vor. Aus diesem Grund assoziiert das Modell nun Wörter, welche auf dem orangenen Einzahlungsschein vorkommen, mit einer Rechnung eines Sportvereins. Dies führt dazu, dass Rechnungen, mit orangem Einzahlungsschein und keinen anderen klaren Indikatoren, fälschlicherweise als Rechnung eines Sportvereins klassifiziert werden.

Diese Problematik lässt sich durch ein sogenanntes Blacklisting der Wörter, welche auf dem orangenen Einzahlungsschein vorkommen, lösen. Dabei dürfen aber nicht nur die exakten Wörter herausgefiltert werden. Auch ähnliche Wörter, sprich Wörter mit Schreib- beziehungsweise OCR-Fehlern müssten gefiltert werden.

**Limitierung auf bekannte Wörter:** Das angewendete Verfahren zum Word embedding, basierend auf einem Wörterbuch, funktioniert nur für bekannte Wörter. Wörter mit einem Tippfehler oder Synonyme können aufgrund dem fehlenden Embedding nicht zur Klassifizierung beitragen.

Anstelle des aktuell verwendeten, einfachen Word Tokenizer könnte ein komplexeres Word embedding (vgl. Kapitel 3.8) angewendet werden. Wie bereits im Kapitel 3.14 erwähnt, zeigen jüngste Forschungen, dass auch in diesem Bereich dank Transfer Learning mit nur wenig Datensätzen gute Ergebnisse erzielt werden können. Das im Oktober 2018 publizierte BERT Modell verspricht in diesem Bereich eine revolutionäre Verbesserung. Das Modell könnte in diesem Fall als Grundlage für das Transfer Learning dienen (Devlin et al., 2018).

### 4.4.3 Schlussfolgerungen

Der Text-basierte Ansatz erzielt eine erheblich bessere Trefferquote als der Bild-basierte Ansatz. Die Analyse der Vorhersagen des Text-basierten Ansatzes zeigt, dass nur wenige Rechnungen fälschlicherweise als Optiker Rechnungen klassifiziert wurden. Mit sechs respektive fünf fälschlicherweise als Fitness und Sportverein klassifizierten Rechnungen aus dem Testset, ist auch für diese beiden Klassen die Genauigkeit hoch.

Während der Fehleranalyse konnte zudem festgestellt werden, dass sich einige dieser Fehler durch die erwähnten Massnahmen sehr wahrscheinlich beheben lassen.

Werden die erwähnten Optimierungsmassnahmen getroffen, so liefert die Text-basierte Klassifizierung eine gute Grundlage zur automatisierten Verarbeitung von Rechnungen. Bevor allerdings eine abschliessende Aussage zur Anwendung von künstlicher Intelligenz im Rechnungseinreichungsprozess getroffen werden kann, muss der Aspekt der Informationsextraktion analysiert werden.

## 4.5 Teil 2 - Informationsextraktion

Um eine Rechnung verarbeiten zu können, müssen Informationen, wie beispielsweise der Totalbetrag oder der Leistungsbezüger, aus dieser extrahiert werden. Eine Lösung zu dieser Problematik der Informationsextraktion wird bereits von vielen existieren Softwarelösungen zur automatisierten Verarbeitung von Rechnungen angeboten. Solche Systeme extrahieren zwei verschiedene Typen von Informationen: Informationen mit Schlüsselwörter und Informationen aus Tabellen (Hamza, Belaïd & Belaïd, 2007).

Zur Extraktion von Informationen aus Tabellen sind viele Ansätze in der Literatur zu finden. Ein einfacher Ansatz von Mandal, Chowdhury, Das und Chanda (2006) erreicht bereits eine Trefferquote von 97.21% (Mandal et al., 2006 in Hamza et al., 2007).

Hamza et al. (2007) stellen eine Lösung zur Informationsextraktion aus Rechnungen vor, welche mit Hilfe von Case Based Reasoning<sup>20</sup>, eine Trefferquote von 76-85% erreicht. Knapp die Hälfte der Fehler wird dabei durch OCR Fehler verursacht.

Im Gegensatz zu den diskutierten Lösungen ist in diesem Experiment die Extraktion von Informationen aus Tabellen nicht relevant. Für die Verarbeitung der Rechnungen im vorliegenden Fallbeispiel ist die Extraktion einzelner Rechnungspositionen ebenfalls nicht relevant.

---

<sup>20</sup>Unter Case Based Reasoning, kurz CBR, werden Ansätze verstanden, bei welchen Entscheidungen aufgrund vergangener Erfahrungen getroffen werden. Die Theorie des Case Based Reasoning sieht das Treffen von Entscheidungen aufgrund von Analogien als zentralen Aspekt der menschlichen Intelligenz und versucht diesen in die künstliche Intelligenz einzubringen (Kolodner, 1999).

In den folgenden Kapiteln werden zwei Lösungsvorschläge zur Extraktion von Informationen aus Rechnungen diskutiert. Die gewählten Ansätze sind, verglichen zu den oben erwähnten Ansätzen von Mandal et al. (2006) und Hamza et al. (2007), einfach gehalten.

Der erste Ansatz stammt aus einem Joint Venture zwischen der AXA, der 3AP AG und der Fachhochschule Nordwestschweiz. Der zweite Ansatz wurde im Rahmen dieser Arbeit erarbeitet.

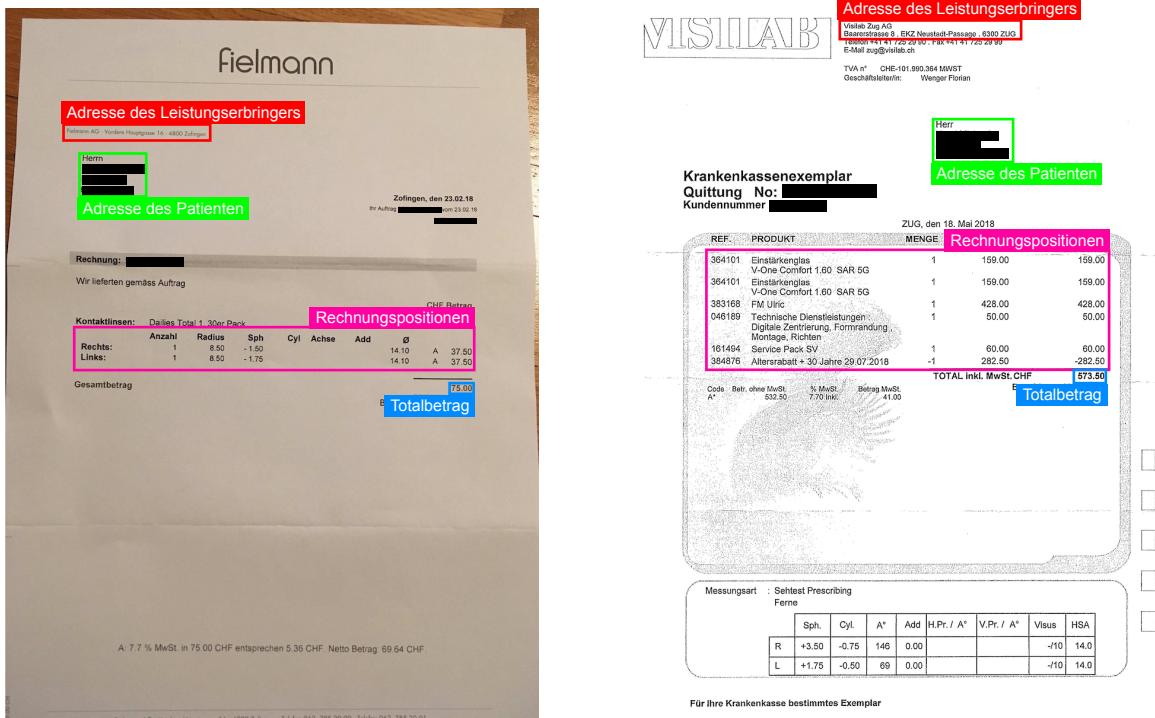
### 4.5.1 Bild-basierte Informationsextraktion

In diesem Kapitel wird ein bild-basierter Ansatz zur Informationsextraktion aus Rechnungen vorgestellt. Der Ansatz stammt aus einem Joint Ventre zwischen der AXA, der 3AP AG und der Fachhochschule Nordwestschweiz.

Auch dieser bild-basierte Ansatz basiert, analog dem bild-basierten Ansatz zur Klassifizierung, auf Algorithmen und Modellen aus dem Bereich der Computer-Vision.

Der präsentierte Ansatz basiert auf der Idee, Modelle, welche ursprünglich zur Objekterkennung in Bildern entwickelt wurden, zur Erkennung von Regionen mit relevanten Informationen (Region of Interest) zu verwenden. Konkret bedeutet dies, dass mit einem Modell zur Objekterkennung die Positionen der Adresse des Patienten, des Namens des Leistungserbringens, der Rechnungspositionen sowie des Totalbetrages auf den Rechnungen identifiziert werden sollen. Die Abbildung 34 zeigt zwei Rechnungen, welche manuell mit den erwarteten Regions of Interest annotiert wurden. Dabei wurden persönliche Daten aus Datenschutzgründen unkenntlich gemacht.

Abbildung 34: Zwei Beispiele von Rechnungen mit annotierten Regions of Interest



(a) Eine über das Kundenportal eingereichte Rechnung

(b) Eine via Post eingereichte Rechnung

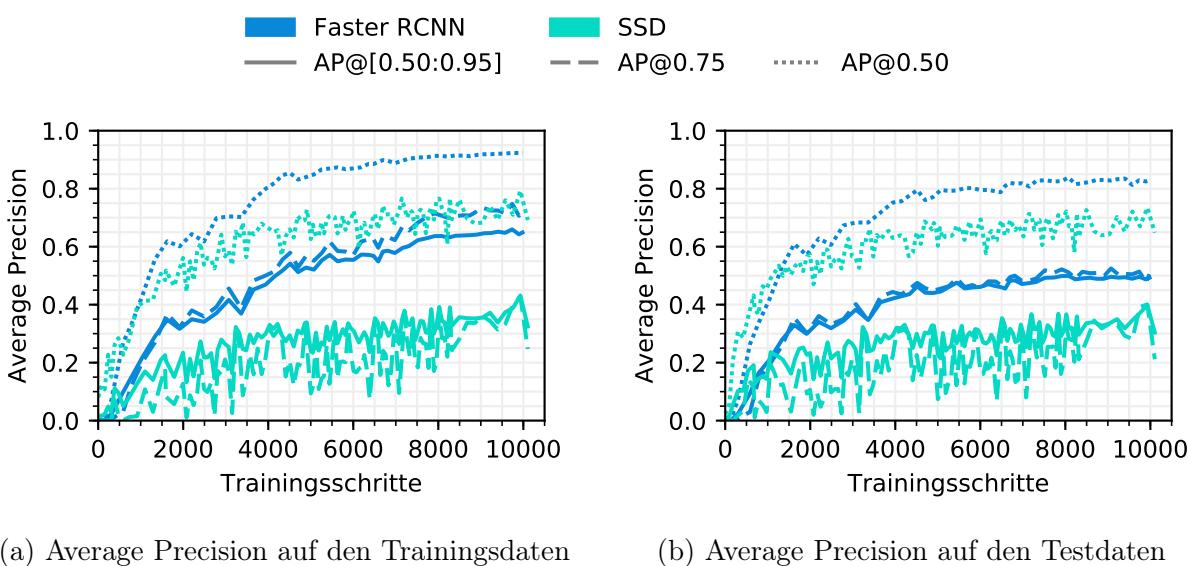
Dieser Ansatz wurde mit dem Framework luminous umgesetzt. Dabei steht das Single-Shot Multibox Detector (SSD) sowie das Faster-RCNN Modell zur Verfügung. Das SSD Modell ist eines der performantesten Modelle zur Objekterkennung. Das Faster-RCNN Modell ist aktuell eines der präzisesten, aber dafür rechenintensivsten, Modelle (Hui, 2018b).

Die beiden Modelle werden auf einem Datensatz von knapp 900 Rechnungen von Optikern trainiert. Diese Rechnungen wurden manuell mit den Regions of Interest annotiert.

Die Abbildungen 35a und 35b zeigen drei verschiedene Mean Average Precision Metriken der beiden Modelle auf den Trainings- respektive Testsdaten. In Abbildung 35b ist zu sehen, dass das Faster-RCNN Modell nach knapp 1'500 Trainingsschritten präziser ist als das SSD Modell. Die Mean Average Precision des Faster-RCNN Modell steigt bei einem IoU Schwellenwert von 0.5 bis auf 0.82. Bei einem IoU Schwellenwert von 0.75 beziehungsweise [0.5 : 0.95] steigt die Mean Average Precision nur knapp über respektive bis knapp unter 0.5. Dies bedeutet, dass die Modelle ungefähr erkennen, wo die relevanten Informationen sind, diese aber nicht genau umrahmen können.

Beim Vergleich der Mean Average Precisions auf den Trainings- und Testdaten ist zu erkennen, dass die Modelle ein hohes Bias aufweisen. Es ist bereits während dem Training eine starke Abweichung von der optimalen Mean Average Precision von 1 zu erkennen. Gegenüber dem Trainingsdatensatz ist die Mean Average Precision auf dem Testdatensatz nur gering kleiner, die Modelle besitzen also eine kleine Varianz. Dies bedeutet, dass die Herkunft der Fehler beim Design der Modelle liegen dürfte und mit mehr Trainingsdaten nicht behoben werden kann.

Abbildung 35: Average Precision auf den Trainings- und Testdaten des Faster-RCNN und SSD Modells



(a) Average Precision auf den Trainingsdaten

(b) Average Precision auf den Testdaten

Abbildung 36 zeigt das Loss der beiden Modelle auf den Testdaten über den Trainingsverlauf hinweg. Es ist zu erkennen, dass das Loss nach 4'000 Trainingsschritten schon sehr flach wird. Die Modelle lernen zu diesem Zeitpunkt also nur noch sehr wenig. Die Modelle noch weiter zu trainieren, würde keine Verbesserung der Mean Average Precision zur Folge haben.

Einen detaillierteren Einblick in die Qualität der Modelle gibt die Tabelle 37. Die Tabelle zeigt die durchschnittliche Intersection over Union der beiden Modelle für die einzelnen zu erkennenden Klassen. Dabei ist festzustellen, dass die beiden Modelle die Adresse des Patienten sowie die Rechnungspositionen relativ gut erkennen können. Bei den anderen beiden Klassen sind die Resultate dagegen ernüchternd.

Auffällig bei der Gegenüberstellung der Mean IoU Werte ist die Klasse Totalbetrag. Dabei schneidet das SSD Modell extrem schlecht ab. Das SSD Modell scheint mit diesen kleinen Regions of Interest besonders Probleme zu haben.

Abbildung 36: Totales Loss auf den Testdaten

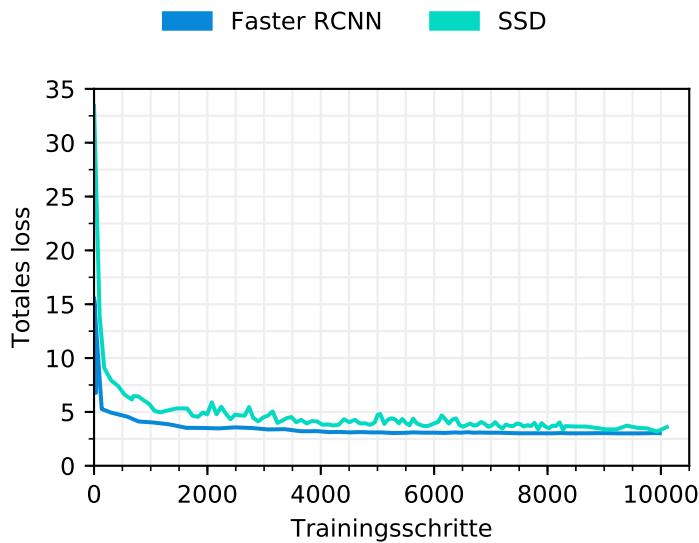


Tabelle 37: Durchschnittliche Intersection over Union der einzelnen Klassen

Klasse	Modell	
	Faster-RCNN	SSD
Adresse des Patienten	0.850	0.731
Adresse des Leistungserbringens	0.625	0.623
Rechnungspositionen	0.835	0.728
Totalbetrag	0.629	0.258

Subsummierend kann festgehalten werden, dass das Faster-RCNN Modell erwartungsgemäss präzisere Vorhersagen macht.

In Bezug auf die Automatisierung der Rechnungseinreichung sind die Adresse des Patienten sowie das Total der Rechnung relevant. Das präsentierte Modell kann aber nur eine der beiden Informationen mit hoher Genauigkeit erkennen. Weiter fehlen andere relevante Informationen wie das Datum des Bezuges der Leistung sowie die Aussage, ob eine ärztliche Verordnung vorliegt oder nicht.

Im folgenden Kapitel wird versucht diesen Ansatz um die fehlenden Informationen zu ergänzen sowie die Qualität der Ergebnisse zu steigern, indem das Modell auf Rechnungen bestimmter Leistungserbringer spezialisiert wird.

#### 4.5.2 Bild-basierte Informationsextraktion pro Rechnungstyp

In diesem Kapitel wird der bild-basierte Ansatz zur Informationsextraktion, welcher im vorherigen Kapitel beschrieben wurde, auf Rechnungen eines einzelnen Leistungserbringens (beispielsweise Fielmann) angewendet. Dadurch soll ein besseres Resultat erzielt werden. Des Weiteren werden die zu erkennenden Regions of Interest so angepasst, dass sich diese mit den Anforderungen aus dem Kapitel 4.2 decken. Das bedeutet, dass neu die Positionen der Klassen Adresse des Patienten, Adresse des Leistungserbringens, Datum des Leistungsbezuges, Totalbetrag und Begründung des Leistungsbezuges (mögliche ärztliche Verschreibung) erkannt werden sollen.

Damit in der Praxis spezifische Modelle zur Informationsextraktion auf die Rechnungen der einzelnen Leistungserbringer angewendet werden können, muss das Modell zur Klassifizierung der Rechnung um die Klassen dieser Leistungserbringer erweitert werden. Nachfolgend wird diese Erweiterung des Modells zur Klassifizierung sowie die Erstellung des Leistungserbringerspezifischen bild-basierten Modell zur Informationsextraktion beschrieben und die Resultate daraus diskutiert.

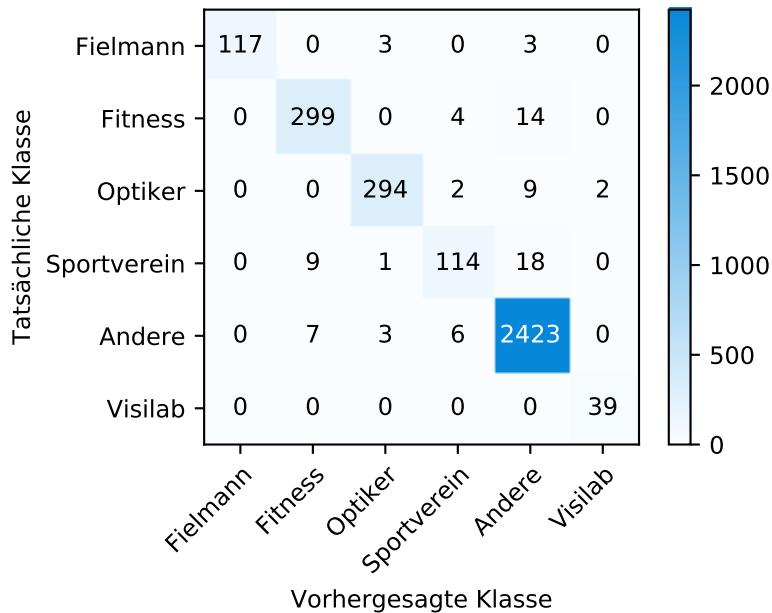
Das im Kapitel 4.4.2 beschriebene, text-basierte Modell zur Klassifizierung der Rechnungen wurde um die Klassen Fielmann und Visilab erweitert. Diese Klassen wurden gewählt, da sie innerhalb der aktuell grössten Klasse von Rechnungen, der Klasse Optiker, einen grossen Anteil haben. Es stehen 578 Rechnungen von Fielmann sowie 195 Rechnungen von Visilab zur Verfügung.

Die Abbildung 38 zeigt die Confusion Matrix des angepassten Modells auf den Testdaten. Darauf ist zu erkennen, dass die Einteilung in die beiden neuen Klassen sehr gut funktioniert. Nach 45 Epochen Training werden innerhalb des Testdatensatzes lediglich 2 Rechnungen fälschlicherweise als Rechnungen von Visilab klassifiziert. Diese Klassifizierung ist nicht ganz falsch, denn es handelt sich tatsächlich um Rechnungen von Visilab, aber in einem anderen Format. Es handelt sich um ein Kassenbeleg sowie eine Monatsrechnung. Diese Unterschiede im Format würden nicht in das erwartete Muster des im zweiten Schritt folgenden Modell zur Informationsextraktion passen und werden deshalb als falsch angesehen.

Die Klassifizierung der Rechnungen nach einzelnen Leistungserbringer scheint für das Modell kein Problem darzustellen. Im Folgenden wird nun die Erkennung der Regions of Interest mit dem bild-basierten Ansatz evaluiert.

Das Faster-RCNN und SSD Modell wurden auf den Rechnungen der Klassen Fielmann und Visilab trainiert. Die Abbildung 39 zeigt die Mean Average Precisions der beiden Modelle für verschiedene IoU Schwellenwerte für die beiden Klassen.

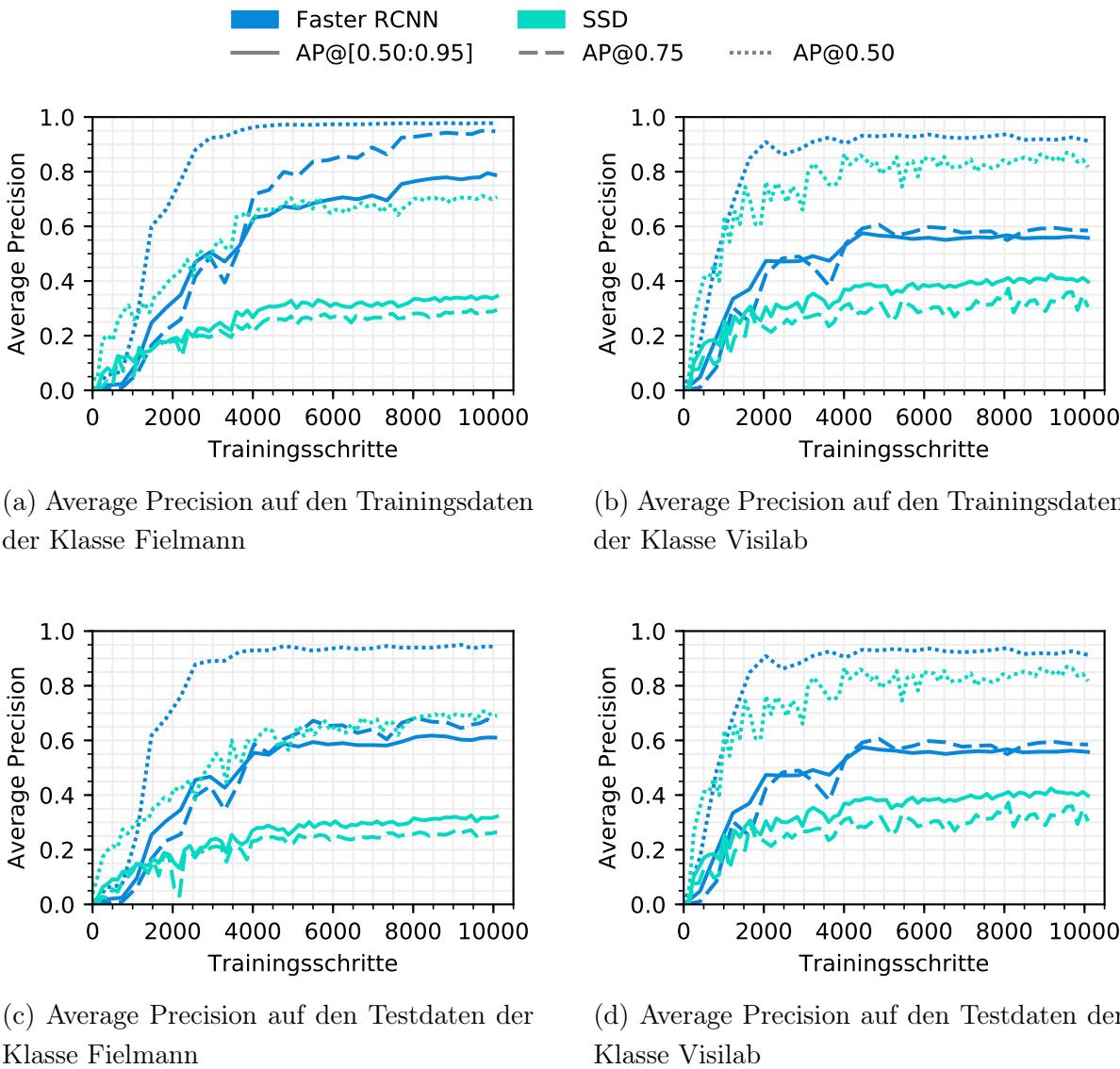
Abbildung 38: Confusion Matrix nach 45 Trainingsepochen des erweiterten Textbasierten Modells zur Klassifizierung von Rechnungen



Die Abbildung 39 zeigt, dass bei den Leistungserbringer spezifischen Modellen der Unterschied in den Mean Average Precisions zwischen dem SSD und Faster-RCNN noch stärker ausfallen (vgl. Abbildung 35).

Die Abbildungen 39a und 39c zeigen die Mean Average Precision des Modells zur Informatonsextraktion aus Rechnungen von Fielmann. Das Modell erreicht während dem Training und dem Testen eine beachtliche Mean Average Precision von 98% respektive 94% bei einem IoU Schwellwert von 0.5. Das Modell erkennt also für sehr viele Rechnungen, wo sich die Regions of Interest befinden. Die mAP@[0.50:0.95] von 80% während dem Training zeigt, dass das Modell die Regions relativ genau lokalisiert. Auf dem Testdatensatz liegt die mAP@[0.5:0.95] dagegen bei nur knapp 60%. Das Modell weist somit bei der mAP@0.5 und mAP@[0.5:0.95] eine Varianz von 4% respektive 20% auf. Aus dieser teilweise grossen Varianz ist zu schliessen, dass das Modell mehr Trainingsdaten bedarf, welche jedoch zum aktuellen Zeitpunkt nicht zur Verfügung stehen.

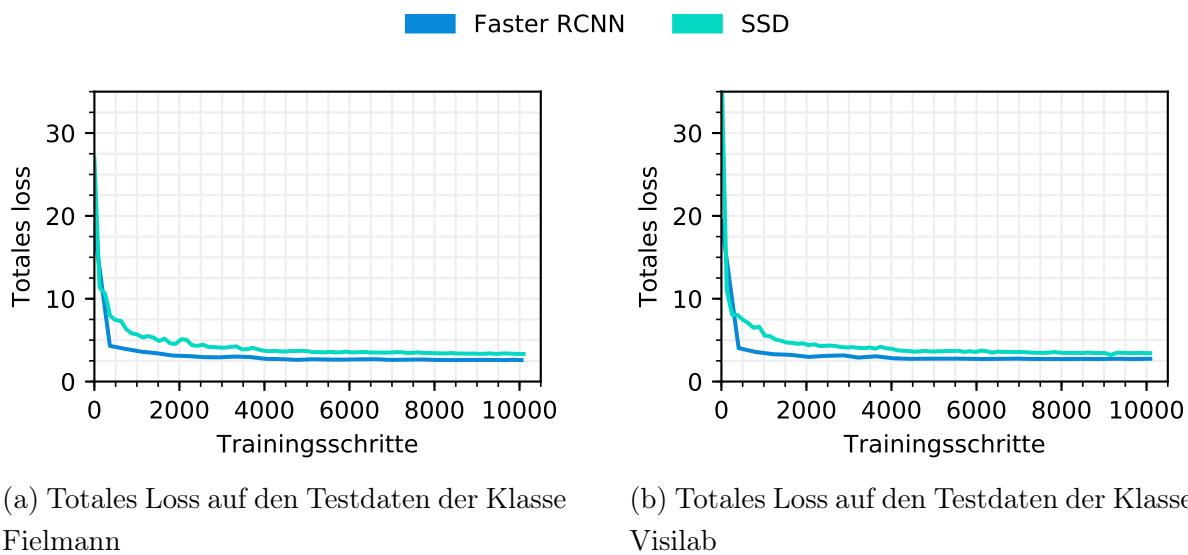
Abbildung 39: Average Precision auf den Trainings- und Testdaten der Leistungserbringer spezifischen Modelle



Der Bedarf an mehr Trainingsdaten zeigt, dass ein solcher Ansatz nicht für alle Leistungserbringer realistisch ist. Für viele Leistungserbringer stehen weniger Trainingsdaten zur Verfügung. Diese Problematik manifestiert sich auch bei der Erkennung der Regions of Interest für Rechnungen von Visilab. Mit 195 Rechnungen stehen hier nur rund ein Drittel so viele Rechnungen wie von Fielmann zur Verfügung. Wie in den Abbildungen 39b und 39d zu sehen ist, liegt die Varianz der  $mAP@[0.5:0.95]$  des Faster-RCNN Modells für Rechnungen von Visilab bei knapp 40%. Dies ist ein klarer Indikator dafür, dass zu wenige Trainingsdaten zur Verfügung stehen.

Die Abbildungen 40a und 40b zeigen das Loss der beiden Modelle auf den Testdaten für beide Klassen. Es ist zu erkennen, dass die Modelle auf den Leistungserbringer spezifischen Rechnungen nach ungefähr 4'000 Trainingsschritten nicht mehr lernen. Die Modelle auf den vorliegenden Trainingsdaten weiter zu trainieren, dürfte auch hier keine Erhöhung der Mean Average Precisions mehr zur Folge haben.

Abbildung 40: Totales Loss auf den Testdaten der Leistungserbringer spezifischen Modelle



Die Tabelle 41 zeigt die durchschnittliche Intersection over Union für die einzelnen Klassen. Es ist zu erkennen, dass das Faster-RCNN Modell deutlich bessere Resultate liefert als das SSD Modell. Erwähnenswert ist, dass das SSD Modell für die Klasse Behandlungsdatum der Rechnungen von Fielmann überhaupt keine und für den Totalbetrag der Visilab Klasse nur sehr schlechte Regions of Interest vorhersagt. Bei diesen beiden Fällen handelt es sich um kleine Regions mit anderen Regions in unmittelbarer Nähe.

Die Klasse Begründung des Leistungsbezugs ist für die Rechnungen von Visilab nicht relevant, da diese Information nicht auf der Rechnung ersichtlich ist.

Da die Leistungserbringer spezifischen Modelle mit anderen Trainings- und Testdaten als die generellen Modell trainiert beziehungsweise getestet wurden, ist ein direkter Vergleich der durchschnittlichen IoU ungenau. Mit Rücksichtnahme auf diese Ungenauigkeit kann aber dennoch festgestellt werden, dass die durchschnittliche IoU der Klasse Totalbetrag bei den Rechnungen von Fielmann um 0.2 grösser ist, als jene über alle Leistungserbringer hinweg (vgl. Tabelle 37 und 41). Bei den Rechnungen von Visilab ist die durchschnittliche IoU um 0.1 grösser als über alle Leistungserbringer hinweg. Dieser mit etwas Ungenauigkeit behafte Vergleich lässt vermuten, dass die Leistungserbringer spezifischen Modelle um einiges präzisere Vorhersagen ermöglichen.

Tabelle 41: Durchschnittliche Intersection over Union der Regions of Interest pro Klasse

Klasse	Modell			
	Fielmann		Visilab	
	Faster-RCNN	SSD	Faster-RCNN	SSD
Adresse des Patienten	0.915	0.884	0.895	0.832
Adresse des Leistungserbringens	0.820	0.692	0.804	0.652
Begründung des Leistungsbezuges	0.864	0.716	-	-
Totalbetrag	0.829	0.672	0.710	0.283
Behandlungsdatum	0.833	0	0.791	0.743

Analog dem Text-basierten Modell zur Klassifizierung wird auch hier eine Fehleranalyse durchgeführt. Die Fehleranalyse der Resultate aus der Informationsextraktion zeigt, dass vor allem Bilder mit schlechter Qualität oder sehr kleinen Regions of Interest zu Ungenauigkeiten führen.

Modelle zur Objekterkennung sind dafür bekannt, dass sie mit kleinen Objekten Probleme haben. Im aktuellen Vorgehen wurden die Bilder auf eine maximale Auflösung von 400x600 Pixeln verkleinert, damit die Rechenintensität tief gehalten werden kann. Dies hat sich beim SSD Modell besonders beim Behandlungsdatum bei Rechnungen von Fielmann gezeigt, welche überhaupt nicht erkannt werden konnten. Hui (2018b) vergleicht verschiedene Modelle zur Objekterkennung mit unterschiedlichen Parametern. Dabei ist festzustellen, dass die Reduktion der Auflösung um die Hälfte in beide Dimensionen (Höhe und Breite) im Durchschnitt eine Reduktion der Trefferquote von ungefähr 15% zur Folge hat. Daraus kann geschlossen werden, dass eine **Erhöhung der Auflösung** die Trefferquote verbessern wird.

Durch die Verkleinerung der Bilder entstehen in einigen Fällen Bilder mit sehr schlechter Qualität. Wird ein anderer Algorithmus zur Verkleinerung der Bilder gewählt, kann die **Qualität der Bilder** verbessert werden. Dadurch sollte das Modell die Strukturen auf den Bildern besser erkennen können und somit genauere Resultate liefern.

Eine Problematik, welche bereits bei der Diskussion der Varianz der Mean Average Precision erläutert wurde, ist der **Mangel an Testdaten**. Im Rahmen dieser Arbeit wurden Rechnungen, welche bei der AXA eingreicht wurden, zum Training verwendet. Um ein Modell in Zukunft besser trainieren zu können, ist es denkbar, Trainingsdaten aus anderen Quellen, beispielsweise vom Leistungserbringer, zu beschaffen oder zu erzeugen. Die Erzeugung von Trainingsdaten umfasst dabei die Erschaffung von Trainingsdaten aufgrund von Abänderungen existierender Daten sowie die Erschaffung von komplett neuen, künstlichen Trainingsdaten. Werden Trainingsdaten erzeugt, so ist darauf zu achten, dass diese möglichst nahe an der Realität sind. Diese künstlich erzeugten Daten sollten nur während dem Training und nicht während dem Testen des Modells verwendet werden, damit die Testresultate realitätsnah bleiben.

### 4.5.3 Vervollständigung des präsentierten Ansatzes

Die präsentierten Experimente zeigen, dass die Regions of Interest mit guter Präzision auf vielen Rechnungen ermittelt werden können. Aus diesen Regions of Interest müssen strukturierte Daten gewonnen werden. Dafür müssen die Regions of Interest durch ein OCR System in Text umgewandelt und in eine strukturierte Form gebracht werden. Auch bei diesen Schritten gibt es Fehlerquellen, so kann beispielsweise das OCR System ungenaue Informationen extrahieren.

Bei Informationen wie der Adresse des Patienten oder des Leistungserbringens ist eine Ungenauigkeit des OCR Systems weniger problematisch. Diese Daten können mit den Stammdaten aus dem Kernsystem der AXA abgeglichen werden. Durch diesen Abgleich können diese Informationen mit hoher Wahrscheinlichkeit aus den Regions of Interest extrahiert werden.

Problematischer ist die Situation beispielsweise beim Totalbetrag oder beim Behandlungsdatum. Macht das OCR System hier einen Fehler, ist es schwierig dies automatisch zu erkennen. Eine Kontrolle, um der Kundin oder dem Kunden keine falsche Abrechnung zuzustellen, könnte hier nur manuell durchgeführt werden. Je nach User Experience die der Kundin oder dem Kunden geboten werden soll, bietet sich eine Kontrolle durch Mitarbeitende oder die Kundin oder den Kunden selbst an. Wird die Kundin oder der Kunde nach dem Upload einer Rechnung im Kundenportal direkt nach der Korrektheit der Daten gefragt, so wäre dies wahrscheinlich akzeptabel. Bei Rechnungen, welche per Post eingereicht wurden, ist eine solche User Experience schwieriger rechtfertigen.

### 4.5.4 Ausblick

Dieses Kapitel zeigt weitere Ansätze zur Informationsextraktion aus den Rechnungen. Es bietet sich an, diese Ansätze mit Experimenten zu evaluieren, um zu sehen, ob eine höhere Genauigkeit möglich wäre, als mit dem bild-basierten Ansatz.

Der bereits in der Einleitung des Kapitels 4.5 präsentierte, auf dem Prinzip des Case Based Reasoning basierende Ansatz von Hamza et al. (2007) liefert gute Resultate zur Extraktion von strukturierten Informationen aus Rechnungen. Das Case Based Reasoning verspricht auf bekannten Rechnungen eine Genauigkeit von über 80% und bei unbekannten Rechnungen von mehr als 75%. Dies Präzision des Modells liegt also etwa gleichauf mit den präsentierten Bild-basierten Ansätzen. Das CBR basierte Modell könnte dabei aber das Trainieren von Leistungserbringer spezifischen Modellen vereinfachen, indem das Modell selbst bereits bekannte Formate von Rechnungen erlernt.

Ein weiterer Ansatz ist die Anwendung eines Text-basierten Named Entity Recognition and Classification Systems (vgl. Kapitel 3.11). Es sind diverse Implementationen dieses Ansatzes als freie Software verfügbar. Ein Beispiel ist die Library SpaCy, mit welcher in nur kurzer Zeit ein problemspezifisches NERC Modell trainiert werden kann. Der aufwendige Teil bei diesem Ansatz ist die Beschaffung der Trainingsdaten. Es müssen alle Texte aus den Rechnungen extrahiert und annotiert werden. Eine Herausforderung bei einer solchen Text-basierten Methode sind wahrscheinlich die Resultate aus dem OCR Schritt. Stehen keine qualitativ hochwertigen Texte zur Verfügung, wird es für ein Modell wahrscheinlich schwierig eine genaue Vorhersage zu machen. Diese Vermutung gilt es in einem Experiment zu überprüfen.

Bei allen Ansätzen hat die Qualität der Resultate des OCR Systems einen Einfluss auf die Qualität des gesamten Systems. Aus diesem Grund kann auch hier festgehalten werden, dass Ansätze zur Verbesserung der Resultate aus dem OCR System (vgl. Kapitel 4.4.2) nachgegangen werden sollte.

### 4.5.5 Schlussfolgerungen

Mit den präsentierten Ansätzen konnten die Regions of Interest mit einer guten Genauigkeit vorhergesagt werden. Die Extraktion und Strukturierung der relevanten Informationen wird an gewissen Stellen noch eine Herausforderung darstellen. Die Informationsextraktion mit Hilfe der künstlichen Intelligenz ist auf jeden Fall möglich und sinnvoll.

Die in dieser Arbeit präsentierten Experimente bieten eine gute Grundlage zur Extraktion der relevanten Informationen. Es ist wichtig, die erwähnten Optimierungsmerkmale anzugehen. Dies bedeutet, dass die Qualität und die Auflösung der Bilder, welche durch das Modell verarbeitet werden, verbessert werden müssen. Am wichtigsten ist es, möglichst viele Trainingsdaten zu beschaffen oder zu erzeugen, damit die Leistungserbringer spezifischen Modelle optimal trainiert werden können.

Die Automatisierung bedarf einer Anpassung im Prozess der Rechnungseinreichung. So ist es wichtig, gewisse Qualitätsprüfungen, wie im Kapitel 4.5.3 angesprochen, in den Prozess einfließen zu lassen. Nur durch diese Anpassungen im Prozess kann ein reibungsloser Ablauf garantiert werden.

# 5 | Empfehlungen und Schlussfolgerungen

In diesem Kapitel werden Empfehlungen an die AXA Gesundheitsvorsorge abgegeben, wie bei der Automatisierung der Rechnungseinreichung vorgegangen werden soll. Weiter wird die Forschungsfrage anhand der Theorie und des Fallbeispiels beantwortet.

## 5.1 Empfehlungen an die AXA Gesundheitsvorsorge

Im Rahmen dieser Arbeit konnte sich der Autor das Wissen aneignen, um die Experimente zur Klassifizierung von Rechnungen und Informationsextraktion aus diesen durchzuführen. Mit vertretbarem Aufwand ist es gelungen Erfolge zu erzielen. Die Investition in die künstliche Intelligenz ist vielversprechend.

Die Klassifizierung der Rechnungen mit dem Text-basierten Ansatz hat sehr gut funktioniert. Wird das aufgezeigte Optimierungspotential ausgeschöpft, kann dieser Aspekt zur Automatisierung mit Hilfe der künstlichen Intelligenz erfolgreich abgedeckt werden. Die Text-basierte Klassifizierung wird aktuell aufgrund der höheren Genauigkeit der Bild-basierten vorgezogen. Es gilt allerdings zu evaluieren, ob sich durch das Hinzufügen von immer mehr Leistungserbringern als Klassen die Ausgangslage ändert.

Bei der allgemeinen Informationsextraktion konnten keine zufriedenstellende Ergebnisse erzielt werden. Die Leistungserbringer spezifischen Modelle zur Informationsextraktion konnten dagegen bessere Resultate erzielen. Damit die Fehlerquote besonders tief gehalten werden kann und somit auch möglichst wenig manueller Aufwand während dem Prozess entsteht, wird der Ansatz des Leistungserbinger spezifischen Bild-basierten Modell zur Informationsextraktion empfohlen. Dieser Ansatz hat im Vergleich zum generellen Bild-basierten Modell eine höhere Genauigkeit und ist in der Lage alle notwendigen Informationen zur Automatisierung einer Rechnung von Fielmann und Visilab zu erfassen. Auch für andere Leistungserbringer ist der Ansatz vielversprechend.

Die in den Experimenten erarbeiteten Modelle funktionieren gut, trotzdem ist es wichtig, die Optimierungspotentiale anzugehen, um eine maximale Automatisierungsquote zu erreichen und dabei die Fehlerquote tief zu halten.

Beispielsweise ist die Investition in eine verbesserte Nutzerführung beim Fotografieren einer Rechnung zentral. Somit können die Qualität der digital eingereichten Rechnungen gesteigert und die Fehler des OCR Systems und der Modelle zur Informationsextraktion reduziert werden.

Die Modelle sollen nicht nur in den aktuellen Prozess integriert werden, sondern der ganze Prozess sollte auf die künstliche Intelligenz aufbauen. Nur mit Qualitätskontrollen und manuellen Korrekturen, wo sich das Modell unsicher ist, kann ein reibungsloser Ablauf garantiert werden. Es könnte beispielsweise lohnenswert sein, die Resultate aus der Automatisierung, mit einer Rückfrage, durch die Kundin oder den Kunden prüfen zu lassen.

Aufgrund der beschriebenen Experimente wird der AXA Gesundheitsvorsorge empfohlen, einen inkrementellen Rollout eines Systems zur Automatisierung der Rechnungseinreichung anzustreben. Dabei wird die Anwendung eines Text-basierten Klassifizierungsmodell und Leistungserbringer spezifischen Bild-basierten Modellen empfohlen.

Durch einen inkrementellen Rollout des Systems kann die Time-to-Market sowie das involvierte Risiko klein gehalten werden. Auch hat die AXA Gesundheitsvorsorge dabei die Möglichkeit die Infrastruktur und den Prozess an die Automatisierung schrittweise anzupassen. Durch dieses Vorgehen erhält die AXA Gesundheitsvorsorge schnell Feedback aus der realen Anwendung und kann gegebenenfalls darauf reagieren.

Neben der Einführung der präsentierten Ansätze wird eine Optimierung dieser sowie die Exploration weiterer Ansätze empfohlen. Nur so kann die Genauigkeit und somit die Automatisierung weiter erhöht werden.

## 5.2 Schlussfolgerungen

Jedes Unternehmen möchte langfristig erfolgreich sein. Nach Porter ist dazu ein Wettbewerbsvorteil unabdingbar. Für die Erreichung eines solchen Wettbewerbvorteils spielen neue Technologien eine immer grösse Rolle. Brynjolfsson und McAfee (2017) vergleichen die künstliche Intelligenz mit der Dampfkraft, Elektrizität und dem Verbrennungsmotor und bezeichnen sie als die wichtigste Allzwecktechnologie unserer Zeit.

Einige Unternehmen möchten die künstliche Intelligenz bereits heute anwenden. Technologie-Giganten investieren enorm, etliche Start-Ups werden in diesem Bereich gegründet. Die Motivation zur Anwendung der künstlichen Intelligenz ist hoch. Dies wird in Zukunft auch kleinere Unternehmen dazu zwingen, in Technologien rund um die künstliche Intelligenz zu investieren.

## 5.2 Schlussfolgerungen

---

Mit erfolgreichen Experimenten in der Fallstudie zeigt diese Arbeit, dass ohne grosses Vorwissen und vertretbarem Aufwand bereits künstliche Intelligenz, welche zur Automatisierung von Geschäftsprozessen beiträgt, geschaffen werden kann. Das Fallbeispiel zeigt allerdings auf, dass die künstliche Intelligenz nicht einfach nur in einen Prozess integriert werden kann, sondern dieser darauf ausgerichtet werden muss. Der Investitionsaufwand ist trotz der mittlerweile guten Zugänglichkeit der künstlichen Intelligenz noch immer nicht zu unterschätzen.

Neben der Erstellung der künstlichen Intelligenz ist die Beschaffung von qualitativ hochwertigen Trainingsdaten eine der grössten Herausforderungen. Im Rahmen dieser Arbeit standen knapp 18'000 Rechnungen zur Verfügung, welche mit vertretbarem Aufwand zum Training verwendet werden konnten. Die Annotation zur Objekterkennung von den knapp 800 Rechnungen von Fielmann und Visilab war hingegen sehr aufwendig.

Trotz der hohen Investitionskosten kann die Forschungsfrage „Können Geschäftsprozesse in kleineren Unternehmen durch eigenentwickelte künstliche Intelligenz automatisiert werden?“ mit Ja beantwortet werden. Auch wenn für ein kleineres Unternehmen die Investitionskosten aktuell ziemlich hoch sein dürften, wird sich dies in den nächsten Monaten stark ändern. Die Geschwindigkeit, in welcher sich die Technologien rund um die künstliche Intelligenz entwickeln ist beeindruckend. Projekte wie das Stanford DAWN Projekt werden die künstlichen Intelligenz einfacher nutzbar machen und somit die notwendigen Investitionskosten reduzieren. Die künstliche Intelligenz wird in Zukunft nicht nur helfen Wettbewerbsvorteile zu erarbeiten sondern wird notwendig sein, um wettbewerbsfähig zu sein.

Der Autor empfiehlt allen Unternehmen sich mit der künstlichen Intelligenz vertraut zu machen und Investitionen zu prüfen, um künftig wettbewerbsfähig zu bleiben.



## 6 | Ausblick

Wie das erarbeitete Fallbeispiel und die Literatur zeigen, sind die grundlegenden Konzepte und Technologien rund um die künstliche Intelligenz bereits auf einem hohen Niveau. Bedarf nach mehr Forschung und Entwicklung gibt es nicht bei den Algorithmen selber, sondern bei deren Nutzbarkeit. Initiativen wie das DAWN Projekt der Stanford University oder [www.fast.ai](http://www.fast.ai) sind erste Schritte, in die Richtung einer einfacher nutzbaren und zugänglicheren künstlichen Intelligenz. Es wird in Zukunft zentral sein, die künstliche Intelligenz nicht nur Spezialistinnen und Spezialisten verfügbar, sondern sie zu einer Allerweltstechnologie zu machen.

Das in dieser Arbeit empfohlene Vorgehen, ein Modell zur Informationsextraktion pro Leistungserbringer zu trainieren, ist aufwendig. Dieser Aufwand könnte wahrscheinlich durch ein System auf Basis von Case Based Reasoning reduziert werden. Ein solches System würde selbstständig verschiedene Kategorien von Rechnungen erlernen. Dadurch muss nur noch ein einziges Modell trainiert und angewandt werden. Aus diesem Grund ist die Forschung in diese Richtung empfehlenswert.



## 7 | Kritische Reflexion

Die vorliegende Arbeit orientiert sich am Fallbeispiel der AXA Gesundheitsvorsorge, da es dem Autoren wichtig war, einen Bezug zur Praxis zu schaffen. Es kann argumentiert werden, dass ein wissenschaftlicher Beweis zur Beantwortung der Forschungsfrage fehlt. Der Autor sieht die Forschungsfrage dennoch als beantwortet, da das Fallbeispiel ein Beweis dafür ist, dass die künstliche Intelligenz zur Automatisierung von Geschäftsprozessen beitragen kann. Die Aspekte von kleineren Unternehmen und der Eigenentwicklung sieht der Autor ebenfalls als behandelt, da er sich das Wissen im Themengebiet der künstlichen Intelligenz im Rahmen dieser Arbeit selbstständig angeeignet hat.

Durch die Aktualität und den schnellen Wandel der behandelten Themen wurde an gewissen Stellen auf Webseiten und Blogs als Quellen zurückgegriffen. Diese wurden vom Autoren durch weitere Recherche zu den jeweiligen Themen geprüft. Weiter wurde jeweils der Hintergrund des jeweiligen Autoren recherchiert, so dass nach der Meinung des Autoren die Quellen als zitierwürdig gelten.



# 8 | Anhang

## 8.1 Literaturverzeichnis

- Ariño de la Rubia, E. (2017). Benchmarking Predictive Models. Zugriff 19. April 2019 unter <https://blog.dominodatalab.com/benchmarking-predictive-models/>
- BAG. (2016). *Antworten auf häufig gestellte Fragen (FAQ) zur Versichertenkarte*. Bern, Schweiz. Zugriff unter <https://www.bag.admin.ch/dam/bag/de/dokumente/kuv-leistungen/Versichertenkarte/faq-versichertenkarte.pdf.download.pdf>
- Bailis, P., Olukotun, K., Ré, C. & Zaharia, M. (2017). Infrastructure for Usable Machine Learning: The Stanford DAWN Project. Stanford, USA.
- Bengio, Y., Ducharme, R. & Vincent, P. (2001). A Neural Probabilistic Language Model. Cambridge, USA: MIT Press.
- BfS. (2018). Finanzierung. Zugriff 30. Oktober 2018 unter <https://www.bfs.admin.ch/bfs/de/home/statistiken/gesundheit/kosten-finanzierung/finanzierung.html>
- Borthwick, A., Sterling, J., Agichtein, E. & Grishman, R. (1998). NYU: Description of the MENE Named Entity System as used in MUC-7. In *Seventh Message Understanding Conference*. New York, New York: New York University.
- Brynjolfsson, E. & McAfee, A. (2017). The Business of Artificial Intelligence. *Harvard Business Review*, (July 2017).
- Buda, M., Maki, A. & Mazurowski, M. A. (2017). *A systematic study of the class imbalance problem in convolutional neural networks* (Diss., Royal Institute of Technology (KTH)).
- Campbell, M., Hoane, A. J. & Hsu, F.-h. (2002). Deep Blue. *Artificial Intelligence*, 134, 57–83.
- Capaul, R. & Steingruber, D. (2010). *Betriebswirtschaft verstehen* (2. Aufl.). Berlin, Deutschland: Cornelsen Schulverlage.
- Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, 37(1), 51–89.

- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. & Kuksa, P. (2011). Natural Language Processing (almost) from Scratch. *The Journal of Machine Learning Research*, 12, 2493–2537.
- CSS Gruppe. (2018). *Geschäftsbericht 2017*. Luzern, Schweiz.
- Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
- EDI. (2017). Faktenblatt Vergütungssysteme. Zugriff 30. Oktober 2018 unter [https://www.priminfo.admin.ch/downloads/fragen-und-antworten/Fiche%20d%20informationtiers%20payant-tiers%20garant\\_DE\\_2018.pdf](https://www.priminfo.admin.ch/downloads/fragen-und-antworten/Fiche%20d%20informationtiers%20payant-tiers%20garant_DE_2018.pdf)
- Explosion AI. (o.D.). Industrial-Strength Natural Language Processing. Zugriff 18. November 2018 unter <https://spacy.io>
- Finanzen.ch. (2017). Axa Winterthur will bis 2020 100'000 Kunden in der Zusatzversicherung gewinnen. Zugriff 30. Oktober 2018 unter <https://www.finanzen.ch/nachrichten/finanzplanung/axa-winterthur-will-bis-2020-100000-kunden-in-der-zusatzversicherung-gewinnen-1002138512>
- Forson, E. (2017). Understanding SSD MultiBox — Real-Time Object Detection In Deep Learning. Zugriff 25. April 2019 unter <https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab>
- Fung, V. (2017). An Overview of ResNet and its Variants. Zugriff 9. März 2019 unter <https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035>
- Godoy, D. (2018). Understanding binary cross-entropy / log loss: a visual explanation. Zugriff 10. April 2019 unter <https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>
- Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*. Cambridge, USA: MIT Press.
- Google LLC. (2018). Google Docs: Kostenlos Dokumente online erstellen und bearbeiten. Zugriff 28. November 2018 unter [https://www.google.com/intl/de\\_ch/docs/about/](https://www.google.com/intl/de_ch/docs/about/)
- Grammarly Inc. (2018). Grammarly: Official Site | Free Grammar Checker. Zugriff 28. November 2018 unter <https://www.grammarly.com>
- Hamza, H., Belaïd, Y. & Belaïd, A. (2007). A case-based reasoning approach for invoice structure extraction. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*.
- He, K., Zhang, X., Ren, S. & Sun, J. (2015). Deep Residual Learning for Image Recognition.
- Howard, J. & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification.
- Hui, J. (2018a). mAP (mean Average Precision) for Object Detection. Zugriff 13. April 2019 unter [https://medium.com/@jonathan\\_hui/map-mean-average-precision-for-object-detection-45c121a31173](https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173)

- Hui, J. (2018b). Object detection: speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3). Zugriff 12. April 2019 unter [https://medium.com/@jonathan\\_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359](https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359)
- Karpathy, A. (2015a). Convolutional Neural Networks (CNNs / ConvNets). Zugriff 25. April 2019 unter <http://cs231n.github.io/convolutional-networks/>
- Karpathy, A. (2015b). Neural Networks Part 1: Setting up the Architecture. Zugriff 28. April 2019 unter <http://cs231n.github.io/neural-networks-1/>
- Kergaßner, R. (2012). IT-Automatisierung schafft Freiräume. *Wirtschaftsinformatik & Management*, 6, 20–23.
- Kha, L. (2000). *Critical Success Factors for Business-to-Consumer E-business: Lessons from Amazon and Dell Le Kha* (Diss., Massachusetts Institute of Technology).
- Kirchgässner, G. (2009). Das schweizerische Gesundheitswesen: Kostenentwicklung. *Die Volkswirtschaft. Das Magazin für Wirtschaftspolitik*, (11/2019), 4–8.
- Kolodner, J. (1999). Instructional Design: Case-Based Reasoning.
- Krogh, A. (2008). What are artificial neural networks? *Nature Biotechnology*, 26(2), 195.
- LanguageTool. (2018). LanguageTool - Prüfung für Rechtschreibung und Grammatik. Zugriff 28. November 2018 unter <https://languagetool.org/de/>
- Le, Q. & Zoph, B. (2017). Using Machine Learning to Explore Neural Network Architecture. Zugriff 9. März 2019 unter <https://ai.googleblog.com/2017/05/using-machine-learning-to-explore.html>
- Lombriser, R. & Abplanalp, P. A. (2015). *Strategisches Management* (6. Aufl.). Zürich, Schweiz: Versus.
- Lu, H., Li, Y., Chen, M., Kim, H. & Serikawa, S. (2018). Brain Intelligence: Go beyond Artificial Intelligence. *Mobile Networks and Applications*.
- Mandal, S., Chowdhury, S. P., Das, A. K. & Chanda, B. (2006). A simple and effective table detection system from document images. *International Journal on Document Analysis and Recognition*, 8(2-3), 172–182.
- Marr, B. (o.D. a). Infervision: Using AI And Deep Learning To Diagnose Cancer. Zugriff 10. April 2019 unter <https://www.bernardmarr.com/default.asp?contentID=1269>
- Marr, B. (o.D. b). The Incredible Ways John Deere Is Using Artificial Intelligence To Transform Farming. Zugriff 10. April 2019 unter <https://www.bernardmarr.com/default.asp?contentID=1387>
- McKinsey Global Institute. (2017). *Artificial Intelligence the next digital frontier?*
- Microsoft Corporation. (2018). Microsoft Word – Textverarbeitungssoftware | Office. Zugriff 28. November 2018 unter <https://products.office.com/de-ch/word>

- Mugan, J. (o.D.). Evaluation and Comparison. Zugriff 18. November 2018 unter <http://www.deepgrammar.com/evaluation>
- Mugan, J. (2018). Persönliche Kommunikation per E-Mail.
- Nadeau, D. & Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1), 3–26.
- Neuberg, B. (2017). Creating a Modern OCR Pipeline Using Computer Vision and Deep Learning. Zugriff 17. November 2018 unter <https://blogs.dropbox.com/tech/2017/04/creating-a-modern-ocr-pipeline-using-computer-vision-and-deep-learning/>
- Ng, A. (2018). Machine Learning Yearning.
- Nielsen, M. (2018). Using neural nets to recognize handwritten digits. Zugriff 25. April 2019 unter <http://neuralnetworksanddeeplearning.com/chap1.html>
- Olah, C. (2014). Deep Learning, NLP, and Representations. Zugriff 28. November 2018 unter <http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/>
- Olah, C. (2015). Understanding LSTM Networks. Zugriff 28. November 2018 unter <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- O.V. (2016). TrainingTesseract 4.00. Zugriff 9. März 2019 unter <https://github.com/tesseract-ocr/tesseract/wiki/TrainingTesseract-4.00>
- O.V. (2018a). 4.0 Accuracy and Performance. Zugriff 17. November 2018 unter <https://github.com/tesseract-ocr/tesseract/wiki/4.0-Accuracy-and-Performance>
- O.V. (2018b). 4.0 with LSTM. Zugriff 17. November 2018 unter <https://github.com/tesseract-ocr/tesseract/wiki/4.0-with-LSTM>
- O.V. (2019). ImproveQuality. Zugriff 11. April 2019 unter <https://github.com/tesseract-ocr/tesseract/wiki/ImproveQuality>
- Patrício, D. I. & Rieder, R. (2018). Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and Electronics in Agriculture*, 153, 69–81.
- Ping An Technology. (o.D.). Ping An Technology. Zugriff 10. April 2019 unter <https://tech.pingan.com/en/>
- Ransbotham, S., Kiron, D., Gerbert, P. & Reeves, M. (2017). *Reshaping Business With Artificial Intelligence: Closing the Gap Between Ambition and Action*.
- Román Aragay, V. (2018). How To Develop a Machine Learning Model From Scratch. Zugriff 19. April 2019 unter <https://towardsdatascience.com/machine-learning-general-process-8f1b510bd8af>
- Rosebrock, A. (2016). Intersection over Union (IoU) for object detection. Zugriff 13. April 2019 unter <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>

- Russell, S. J. & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach* (3. Aufl.). New Jersey, USA: Prentice Hall.
- Sarter, N. B., Woods, D. D. & Billings, C. E. (1997). Phase-change chalcogenide nonvolatile ram completely based on CMOS technology. *Handbook of Human Factors & Ergonomics*, (2), 29–31.
- Scheidl, H. (2018). An Intuitive Explanation of Connectionist Temporal Classification. Zugriff 28. November 2018 unter <https://towardsdatascience.com/intuitively-understanding-connectionist-temporal-classification-3797e43a86c>
- Shung, K. P. (2018). Accuracy, Precision, Recall or F1? Zugriff 14. März 2019 unter <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>
- Smith, R. (2007). An overview of the tesseract OCR engine. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*.
- Sorio, E., Bartoli, A., Davanzo, G. & Medvet, E. (2012). A Domain Knowledge-based Approach for Automatic Correction of Printed Invoices. *International Conference on Information Society (i-Society 2012) For*, 151–155.
- Stanford NLP Group. (o.D.). Stanford Named Entity Recognizer (NER). Zugriff 18. November 2018 unter <https://nlp.stanford.edu/software/CRF-NER.shtml>
- The Economist. (2010). Tech.View: Cars and software bugs. Zugriff 9. März 2019 unter <https://www.economist.com/babbage/2010/05/16/techview-cars-and-software-bugs>
- The Economist. (2018). The sunny and the dark side of AI. Zugriff 25. November 2018 unter <https://www.economist.com/special-report/2018/03/28/the-sunny-and-the-dark-side-of-ai>
- Tredinnick, L. (2017). Artificial intelligence and professional roles. *Business Information Review*, 34, 37–41.
- Tsang, S. (2018). Review: Inception-v4 — Evolved From GoogLeNet, Merged with ResNet Idea (Image Classification). Zugriff 9. März 2019 unter <https://towardsdatascience.com/review-inception-v4-evolved-from-googlenet-merged-with-resnet-idea-image-classification-5e8c339d18bc>
- Turian, J., Ratinov, L. & Bengio, Y. (2010). *Word representations: A simple and general method for semi-supervised learning*. Zugriff unter <http://metaoptimize>.
- Uettwiller-Geiger, D. (2005). A lab's strategy to reduce errors depends on automation. *Medical Laboratory Observer*, 37(12), 26. Zugriff unter [http://www.mlo-online.com/articles/1205/1205lab\\_mgmt.pdf](http://www.mlo-online.com/articles/1205/1205lab_mgmt.pdf)
- van der Maaten, L. & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605.
- van Rijsbergen, C. J. (1979). Information Retrieval. Glasgow, Schottland: University of Glasgow.

- Volk, M. (o.D.). Formale Grammatiken und Syntaxanalyse: Strukturelle Mehrdeutigkeiten. Zugriff 28. November 2018 unter [https://files.ifi.uzh.ch/cl/volk/SyntaxVorl/Vorl\\_10.Ambig.html](https://files.ifi.uzh.ch/cl/volk/SyntaxVorl/Vorl_10.Ambig.html)
- Weiss, T. (2016). Deep Spelling. Zugriff 18. November 2018 unter <https://machinelearnings.co/deep-spelling-9ffef96a24f6>
- Xiao, L. (2004). *Information extraction in the practical applications* (Diss., Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)).
- Zoph, B., Vasudevan, V., Shlens, J. & Le, Q. (2017). AutoML for large scale image classification and object detection. Zugriff 9. März 2019 unter <https://ai.googleblog.com/2017/11/automl-for-large-scale-image.html>

## 8.2 Tabellen- und Abbildungsverzeichnis

1	Wettbewerbsstrategien nach Porter . . . . .	5
2	Modell eines Neurons nach McCulloch und Pitts . . . . .	12
3	Modell eines neuronalen Netzwerks mit zwei Fully Connected Layer . . . . .	14
4	Konzept der linearen Separierbarkeit . . . . .	15
5	Modell eines tiefen neuronalen Netzwerks mit einer versteckten Schicht . . . . .	15
6	Verhältnis der verfügbaren Daten und der Genauigkeit unterschiedlich grosser neuronaler Netze . . . . .	16
7	Over- und Underfitting dargestellt anhand von Graphen von Funktionen . . . . .	18
8	Visualisierung eins Convolution Layer . . . . .	21
9	Informationsfluss durch ein Recurrent Neural Network . . . . .	23
10	Informationsfluss eines LSTM Netzwerk . . . . .	24
11	Modulares Netzwerk zur Validierung von 5-Grammen . . . . .	25
12	t-SNE Darstellung eines Word embeddings . . . . .	26
13	Sechs Ausgangswörter mit den ihnen ähnlichssten Word embeddings . . . . .	27
14	Vergleich der Erfolgsrate bei der Prüfung von 418 Textsnippets . . . . .	28
15	Vergleich des NASNet mit anderen Netzwerken zur Klassifizierung von Bildern . . . . .	32
16	Resultate aus einem Modell zur Objekterkennung in Bildern . . . . .	33
17	Beispiel einer Confusion Matrix . . . . .	35
18	Elemente zur Berechnung der Genauigkeit in einer Confusion Matrix . . . . .	36
19	Elemente zur Berechnung der Sensitivität in einer Confusion Matrix . . . . .	37
20	Beispiel einer tatsächlichen und vorhergesagten Position eines Objekts . . . . .	39
21	Beispiele des Intersection over Union Mass . . . . .	39
22	Beispiel der Berechnung der Genauigkeit und Sensitivität . . . . .	40
23	Beispiel einer PR-Curve . . . . .	41
24	Beispiel zur Glättung einer PR-Curve . . . . .	41
25	Interpolation einer geglätteten PR-Curve. . . . .	42
26	Vergütungsmodelle bei den Schweizer Krankenversicherern . . . . .	48
27	Prozess der Rechnungseinreichung und -verarbeitung der AXA Gesundheitsvorsorge . . . . .	51
28	Ungleichverteilung der Klassen innerhalb des Trainingsdatensatzes . . . . .	56
29	neuronale Netze, welche bei der Bild-basierten Klassifizierung zur Anwendung kommen . . . . .	57
30	Statistiken aus dem Training der Bild-basierten Klassifizierung von Rechnungen . . . . .	58
31	Neuronales Netzwerk, welches bei der Text-basierten Klassifizierung zur Anwendung kommt . . . . .	60
32	Statistiken aus dem Training der Text-basierten Klassifizierung von Rechnungen . . . . .	61
33	Confusion Matrix nach 23 Trainingsepochen des Text-basierten Modells zur Klassifizierung von Rechnungen . . . . .	62

34	Zwei Beispiele von Rechnungen mit annotierten Regions of Interest . . . . .	68
35	Average Precision auf den Trainings- und Testdaten des Faster-RCNN und SSD Modells . . . . .	69
36	Totales Loss auf den Testdaten . . . . .	70
37	Durchschnittliche Intersection over Union der einzelnen Klassen . . . . .	70
38	Confusion Matrix nach 45 Trainingsepochen des erweiterten Text-basierten Modells zur Klassifizierung von Rechnungen . . . . .	72
39	Average Precision auf den Trainings- und Testdaten der Leistungserbringer spezifischen Modelle . . . . .	73
40	Totales Loss auf den Testdaten der Leistungserbringer spezifischen Modelle .	74
41	Durchschnittliche Intersection over Union der Regions of Interest pro Klasse	75

### 8.3 Sourcecode

Der während dieser Arbeit erarbeitete Sourcecode ist auf der Plattform GitHub, unter dem Link <https://github.com/sventschui/thesis>, zu finden.