Fachhochschule Südwestfalen
University of Applied Sciences

Bachelorarbeit

Automatische Analyse von Stimmmerkmalen zur Vorhersage von Persönlichkeitsprofilen mittels Künstlicher Intelligenz

Sebastian Schmidt

Matrikelnummer: 10053783

E-Mail: schmidt.sebastian2@fh-swf.de

Erstprüfer: Prof. Dr. Michael Rübsam

Zweitprüfer: Prof. Dr. Christian Gawron

25. Oktober 2019

Eigenständigkeitserklärung

Ich erkläre, dass ich die Arbeit selbständig angefertigt und nur die angegebenen Hilfsmittel benutzt habe. Alle Stellen, die dem Wortlaut oder dem Sinn nach anderen Werken, gegebenenfalls auch elektronischen Medien, entnommen sind, sind von mir durch Angabe der Quelle als Entlehnung kenntlich gemacht. Entlehnungen aus dem Internet sind durch Angabe der Quelle und des Zugriffsdatums belegt. Weiterhin habe ich die vorliegende Arbeit an keiner anderen Stelle zur Erlangung eines Abschlusses vorgelegt.

Datum, Ort

Unterschrift

Inhaltsverzeichnis

Abbildungsverzeichnis

Listingverzeichnis

Formelverzeichnis

1. Einleitung

2. Eingrenzung

3. Theoretische Grundlagen

In diesem Abschnitt sollen theoretische Grundlagen des Maschinellen Lernens vorgestellt werden. Es wird zunächst auf allgemeine Grundlagen eingegangen, um im folgenden auf die spezielleren Neuronalen Netze und Konvolutionellen Neuronale Netze einzugehen.

3.1. Allgemeine Grundlagen des Maschinellen Lernens

Maschinelles Lernen gibt einem Computer die Möglichkeit zu lernen ohne explizit von einem Entwickler für eine Aufgabe programmiert zu werden. Tom Mitchell beschreibt Maschinelles Lernen 1997 mit einem Computerprogramm das ohne Veränderungen des Programmcodes aus Erfahrungen E im Bezug auf eine Aufgabe T und ein Maß für die Leistung P lernt, indem seine Leistung P mit der Erfahrung E anwächst [1, S. 4]. Erfahrungen sammelt ein System, indem Daten mit speziellen Algorithmen betrachtet werden. Auf Basis dieser Daten erstellt der Algorithmus dann eine Strukturbeschreibung, welche auch als Modell bezeichnet wird [2, S. 2].

In diesem Projekt wäre die Aufgabe das Klassifizieren von Gebärdensprache. Die Leistung des Modells würde durch die korrekte Klassifizierung beschrieben werden, welche idealerweise während des Trainingsvorgangs beim Anlernen von Datensätzen zu Gebärdensprache ansteigen sollte.

3.1.1. Arten von Verfahren des Maschinellen Lernens

Verfahren des Maschinellen Lernens werden in der Literatur häufig mittels verschiedener Kategorien in verschiedene Arten aufgeteilt. Auch wenn viele Arten von Verfahren in dieser Arbeit nicht betrachtet werden, ist es wichtig diese zu benennen, um ein passendes Verfahren auszuwählen. Es werden typischerweise drei Kategorien betrachtet, die Verfahren einteilen [1, S.8-14][3, S.2].

Zunächst wird ein Verfahren danach eingeteilt, welche Informationen zu den Trainingsdaten anhand von Label nötig sind. Muss jeder Datenpunkt mit einem Label versehen sein, so handelt es sich um Überwachtes Lernen. Dem gegenüber steht das Unüberwachte Lernen, bei dem ein Algorithmus versucht Aussagen über mögliche Label aufgrund der Struktur der Daten zu treffen. Das Halbüberwachte Lernen kombiniert beide Eigenschaften und der Algorithmus versucht zunächst Label anzulernen, um im Folgenden autonom fortzufahren. Abschließend sei noch das Reinforcement Lernen zu nennen. Dieses bestraft und belohnt Aktionen auf eine vorher definierte Weise. Der Algorithmus versucht dann möglichst viele Belohnungen in möglichst wenig Zeit zu erhalten und gleichzeitig Bestrafungen zu verhindern [1, S.8-14][3, S.2].

Eine weitere Kategorie ist die Art wie ein Lernverfahren mit Daten versorgt werden muss. Beim Batch-Learning muss das Verfahren auf Basis eines kompletten Datensatzes trainiert und kann danach nicht mehr mit weiteren Daten verbessert werden. Es muss von Grund auf neu trainieren. Dem gegenüber steht das Online-Learning, welches nach und nach trainiert wird, und noch eine spätere Verfeinerung ermöglicht [1, S.14-17][3, S.2].

Mit einer weiteren Kategorie wird festgelegt wie ein Verfahren Daten verallgemeinert, um neue Aussagen treffen zu können. Mit modellbasierten Lernen wird anhand der Daten ein mathematisches Modell erstellt, welches versucht die Struktur der Daten korrekt zu beschreiben, um neue Vorhersagen treffen zu können. Instanzbasiertes Lernen vergleicht bei vorherzusagenden Daten die Merkmale mit bereits angelernten Datensätzen. Werden hier Ähnlichkeiten in den bereits erlernten Instanzen entdeckt, werden diese für eine neue Aussage genutzt [1, S.14-17][3, S.2].

Ebenfalls lassen sich Verfahren nach der Art des zu lösenden Problems einteilen. Bei der Klassifikation werden Daten anhand ihrer Label einer gewissen Klasse zugeordnet. Das Ziel ist es nun die Klasse von neuen Datensätzen korrekt vorherzusagen. Demgegenüber stehen Regressionsprobleme bei denen ein Label einen konkreten Wert angibt, welcher bei einem neuen Datensatz vom Modell korrekt vorhergesagt werden muss [1, S.8-9][3, S.2].

Bei dem in dieser Arbeit zu lösenden Problem handelt es sich um ein typisches Klassifikationsproblem des Überwachten Lernens. Ein Modell wird auf Basis von vorklassifizierten Daten, also typischerweise Datensätze mit Bildern von Händen die ein Zeichen in Gebärdensprache repräsentieren, angelernt. Dieses soll dann im Folgenden weitere Bilder korrekt einem Zeichen in Gebärdensprache zuweisen können. Instanzbasiertes Lernen könnte bei Bilddaten zu Problemen führen. Da Bilder in der Regel relativ viele Daten beinhalten, könnte ein Merken der Beispiele zu sehr großen Modellen führen [1, S.18]. Aus diesem Grund sollte modellbasiertes Lernen vorgezogen werden. Es ist allerdings sowohl Batch-Learning als auch Online-Learning denkbar.

3.1.2. Typische Probleme beim Maschinellen Lernen

Beim Maschinellen Lernen können viele Probleme auftreten, die den Lernerfolg eines Modells behindern können. Um diesen entgegenwirken zu können, sollten typische Probleme vor dem Entwickeln eines Modells betrachtet werden.

Schon bei einfachen Problemen ist eine ausreichende Datenmengen nötig, um Modelle antrainieren zu können. Abhängig von der Komplexität eines Problems und des genutzten Algorithmus, können Tausende bis Millionen von Datensätzen für einen Lernvorgang nötig sein [1, S.23][3, S.4].

Wurden Daten fehlerhaft gesammelt und repräsentieren nicht das Spektrum der möglichen Eingabe im Einsatz, so kann ein Modell keine korrekten Vorhersagen für den geplanten Einsatz treffen, da hier das Wertespektrum nicht das Gelernte repräsentiert. Auch eine ungünstige Verteilung von Klassen in den Daten, kann zu Probleme führen, besonders wenn dieser Fall mit unpassenden Qualitätsmaßen ausgewertet wird [1, S.24-25][3, S.4].

Minderwertige Daten erschweren ebenfalls den Lernvorgang. Dies kann sich durch verrauschte, fehlerhafte und fehlende Daten sowie Ausreißer äußern. Eine manuelle oder automatische Vorverarbeitung könnte dies verbessern. Sogar ein Auslassen bestimmter Merkmale kann sich als sinnvoll herausstellen. [1, S.26][3, S.3]. Hilft dies nicht ist ein Sammeln von neuen Datensätzen sinnvoll.

Irrelevante oder schlecht gewählte Merkmale für das Antrainieren eines Datensatzes führen ebenfalls zu einem schlechteren oder gar keinem Lernerfolg. Für das Vorhersagen von Gebärdensprache, sollte zum Beispiel das Datum des vorherzusagenden Bildes keinen Einfluss haben. Wird diese Information im Trainingsvorgang trotzdem einbezogen, so ist mit einer schlechteren Performanz des Lernvorgangs zu rechnen [1, S.26][3, S.4].

Overfitting tritt auf, wenn ein Modell für die Trainingsdaten zu komplex gewählt ist. Es führt dazu, dass ein Modell nicht korrekt verallgemeinert, sondern zufällige Variationen in den Trainingsdaten lernt. Es äußert sich in einer guten Performanz im Training, während diese in Tests nachlässt. Verhindern lässt sich dies im Modell durch eine Verringerung der Zahl der Parameter oder bestimmter Techniken, die dem Modell Restriktionen auflegen, der sogenannten Regularisierung. Außerdem lässt es sich durch Sammeln neuer, sowie der Verringerung von Rauschen in vorhandenen, Daten reduzieren [1, S.27][3, S.13].

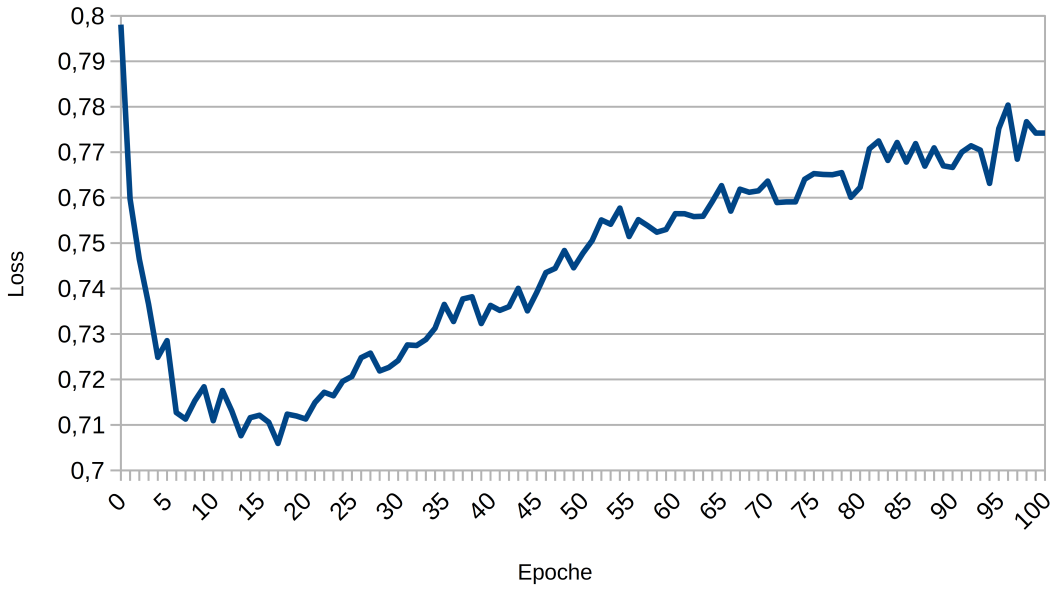

Abbildung 1: Nach der 20. Epoche, tritt in diesem mehrschichtigen Neuronalen Netz ein Overfitting auf [3, S.14].

Zuletzt sei das Underfitting genannt, welches dem Overfitting gegenübersteht. Ein Modell ist dabei zu einfach, um die Strukturen in den Trainingsdaten anzulernen. Modelle mit mehr Parametern, bessere Merkmale oder eine Verringerung der Regularisierung, reduzieren in der Regel ein Overfitting [1, S.29][3, S.14].

3.1.3. Datenvorverarbeitung für Bilddaten

Wie im vorherigen Kapitel dargestellt, würden qualitativ unzureichende Daten zu einer schlechteren Performanz beim Anlernen eines Modells führen. Dies kann auch bei Bilddaten der Fall sein und sollte durch Vorverarbeitungsschritte verhindert werden. In dieser Sektion sollen einige typische Schritte der Bildvorverarbeitung dargestellt werden.

Verrauschte Bilder können bei der Bilderkennung große Probleme darstellen. Ist ein Bild in keiner Weise rauschfrei, wurde gezeigt, dass Verfahren zum Entfernen von Rauschen eine positiven Effekt haben können [4]. Dieses Verfahren wurde im Praxisteil zwar, aufgrund des qualitativ hochwertigen Datensatzes, nicht genutzt,

stellt aber für die Zukunft, besonders im Einsatz wo die Qualität der Bilddaten nicht immer gewährleistet ist, eine gute Methode zu einer möglichen Verbesserung dar.

Ein weiterer Datenvorverarbeitungsschritt ist das Extrahieren von relevanten Bildausschnitten. Die meisten Datensätze für die Erkennung von Gebärdensprache, haben hier bereits die relevanten Ausschnitte vorverarbeitet, sodass nur noch die Hände zu sehen sind. Dies ist jedoch ein Problem für die Praxis, da ein Bild einer Webcam in der Regel mehr als nur die Hand beinhaltet. Es würde hier das Problem entstehen, dass die Daten in der Praxis nur wenig mit dem Angelernten übereinstimmen. Hier ist es sinnvoll den relevanten Teil eines Bildes zu extrahieren. Dazu wurde im Praxisteil ein Machine-Learning Framework names MediaPipe genutzt [5].

Zuletzt müssen Bilder in der Regel vor der Nutzung skaliert werden. Ein Modell des Maschinellen Lernens erwartet in der Regel eine feste Eingabegröße. Dies gilt auch für Bilder, sodass als Beispiel das VGG19 Konvolutionelle Neuronale Netzwerk für Objekterkennung eine Eingabegröße von 224 x 224 Pixeln erwartet [6, S.2]. Auch mit Blick auf die Laufzeit ergibt eine Skalierung von Bildern Sinn. Moderne Webcams erreichen schon häufig eine Auflösung von 1920 x 1080 Pixeln. Dies würde bei unskalierten und nicht extrahierten Bildinformationen zu einer Eingabegröße von 2. Millionen Parametern führen, was abhängig vom Modell zu sehr hohen Laufzeiten führen kann. Typische Algorithmen zum herunterskalieren von Bildern sind die Nächste-Nachbarn, Bilineare und Bikubische Interpolation [6, S.2].

3.1.4. Qualitätsmaße

Um die Leistung eines Modells korrekt bewerten zu können, müssen abhängig vom Ziel Qualitätsmaße eingeführt werden, die es möglich machen Modelle zu vergleichen. Dazu wird mit der Kreuzentropie ein typisches Qualitätsmaß für Klassifikationsprobleme eingeführt. Außerdem werden die Begriffe Konfusionsmatrix, Accuracy, Precision, Recall und F1-Score diskutiert.

In praktisch allen Fällen können Algorithmen des Maschinellen Lernens nicht mit Klassenbezeichnungen in Form von Strings umgehen. Eine Menge von Klassenbezeichnungen wie $\{a, b, c\}$ könnte nur schwierig direkt von Algorithmen als Label genutzt werden. Hier werden in der Regel zwei Möglichkeiten genutzt, um Bezeich-

nungen in nützlichere Label umzuwandeln. Diese wären:

- Klassen-IDs: Die vorher angegebene Menge von Klassenbezeichnungen könnte in eine Menge von repräsentierenden IDs transformiert werden. So wird die Menge $\{a, b, c\}$ in die IDs $\{0, 1, 2\}$ umgewandelt. Der Algorithmus muss bei der Klassifikation nun eine korrekte IDs vorhersagen, welche eine Klasse repräsentiert. Aufgrund der Arbeitsweise vieler Algorithmen nehmen diese allerdings an, dass IDs die näher beieinander liegen ähnlicher zueinander sind. Dies ist nicht für alle Problemstellungen ideal [3, S.12][1, S.63].
- One-Hot-Kodierung: Bei diesem Verfahren wird für jede Klasse eine Zahl in einem Vektor reserviert. Eine vorliegende Klasse wird dann in der Regel mit einer 1 und der Rest der Elemente mit einer 0 markiert. Für die Beispielmenge $\{a, b, c\}$ ergebe sich daher die Menge von Vektoren $\{(1\,0\,0), (0\,1\,0), (0\,0\,1)\}$. Korrekt klassifiziert wurde nun, wenn der vorliegende Klasse mit der höchsten Vorhersagen markiert wird. Da in diesem Fall keine Beziehung zwischen den Klassen erkannt wird, ist dieses Darstellung auch ideal für die Problemstellung [3, S.12][1, S.63].

Die bereits vorher erwähnte Kreuzentropie ist ein ideales Qualitätsmaß für Klassifikationsprobleme in One-Hot-Kodierung. Niedrige Vorhersagen bei der vorliegenden Klassen werden dabei abgestraft, während andere Vorhersagen ignoriert werden. Für einen Datensatz mit m Tupeln, seinen One-Hot-kodierten Labels **y** mit K-Klassen und den One-Hot-kodierten Vorhersagen **y'** mit K-Klassen gilt [3, S.13][1, S.63]:

$$J = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} \mathbf{y}_{ik} \cdot \log(\mathbf{y}'_{ik}) \tag{1}$$

Nun, wurde ein Qualitätsmaß eingeführt mit dem sich die Korrektheit einer Vorhersage bestimmen lassen kann. Im Folgenden werden nun Wege beschrieben, wie sich die Performanz eines Modells auf einem gesamten Datensatz auswerten lässt. Eine typische und ausführliche Auswertungsmöglichkeit bei Klassifikationsproblemen ist eine Konfusionsmatrize, für welche ein Beispiel in Abbildung 2 zu sehen ist. Jede Zeile in einer Konfusionsmatrize steht für eine tatsächliche Klasse, während eine Spalte eine vorhergesagte Kategorie repräsentiert (dies ist definitionsabhängig und kann auch vertauscht werden). Die Wertepunkte in der Matrize beschreiben damit, als welche Klasse ein Datenpunkt einer tatsächlichen Klasse vorhergesagt wird. Sie werden meistens als feste Anzahl oder Anteile in Prozent aller Vor-

hersagen zu dieser Klasse angegeben. Die Hauptdiagonale der Matrize beschreibt alle korrekt klassifizierten Vorhersagen [1, S.86-87].

	A	B	C
A	90	4	6
B	12	85	2
C	4	19	80

Abbildung 2: Beispiel einer Konfusionsmatrize für die Klassen A, B, C.

Eine Konfusionsmatrize ist sehr ausführlich und beinhaltet besonders bei vielen Klassen viele Informationen, die nicht unbedingt für die Lösung des betrachteten Problems relevant sind. In diesem Fall können andere Qualitätsmaße sinnvoll sein. Um diese zu Errechnen ist ein Verständnis über die Begriffe True Positive (TP), True Negative (TN), False Positive (FP) und False Negative (FN) nötig [1, S.87]. Diese sind beispielsweise abhängig zu Klasse A in der Abbildung 3 zu erkennen. Es gilt:

- True Positive (grün): Beschreibt den korrekt klassifizierten Anteil der betrachteten Klasse.
- True Negative (hellgrün): Beschreibt den korrekt klassifizierten Anteil der nicht betrachteten Klassen.
- False Positive (rot): Beschreibt den Anteil mit dem die nicht betrachteten Klassen fälschlicherweise als die Betrachtete eingeordnet werden.
- False Negative (orange): Beschreibt den Anteil mit dem eine betrachtete Klasse fälschlicherweise als die nicht betrachteten Klassen eingeordnet wird.

	A	B	C
A	90	4	6
B	12	85	2
C	4	19	80

Abbildung 3: Beispiel für TP, TN, FN, FP der Klasse A in einer Konfusionsmatrize.

8. Quellen

[1] Aurelien Geron. *Praxiseinstieg: Machine Learning mit Scikit-Learn & TensorFlow*. Heidelberg: O'Reilly, 2018. isbn: 978-3-96009-061-8.

[2] Josh Patterson und Adam Gibson. *Deep Learning: A Practitioner's Approach*. Sebastopol: O'Reilly, 2017. isbn: 978-1-491-91425-0.

[3] Sebastian Schmidt. "Automatische Analyse von Stimmmerkmalen zur Vorhersage von Persönlichkeitsprofilen mittels Künstlicher Intelligenz". Iserlohn: Fachhochschule Südwestfalen, 25. Okt. 2019.

[4] Jonghwa Yim und Kyung-Ah Sohn. "Enhancing the Performance of Convolutional Neural Networkson QualityDegraded Datasets". Suwon, Korea: Department of Computer Engineering: Ajou University, 18. Okt. 2017. url: https://arxiv.org/ftp/arxiv/papers/1710/1710.06805.pdf (besucht am 05. 07. 2020).

[5] Google LLC. *Cross-platform ML solutions made simple*. url: https://google.github.io/mediapipe/ (besucht am 05. 07. 2020).

[6] Erwin Quiring, David Klein, Daniel Arp, Martin Johns und Konrad Rieck. "Adversarial Preprocessing: Understanding and PreventingImage-Scaling Attacks in Machine Learning". Braunschweig, Germany: Technische Universität Braunschweig, 2020. url: https://www.usenix.org/system/files/sec20fall_quiring_prepub.pdf (besucht am 05. 07. 2020).