HTW Berlin
Wintersemester 2019

Masterstudiengang Angewandte Informatik
Forschungsprojekt Teil A

Analyse von Sensordaten in einem Lernkontext mit Hilfe neuronaler Netze

Sascha Witte
Matrikelnummer: s0545027
s0545027@htw-berlin.de

betreut durch
Prof. Dr. Albrecht Fortenbacher

Berlin den 20. April 2019

Inhaltsverzeichnis

Zusammenfassung

Das Forschungsprojekt Teil A beschäftigt sich mit der Implementierung und Evaluation von verschiedenen Neuronalen Netzarchitekturen. Im Vordergrund stehen dabei das Convolutional Neural Network, sowie das Recurrent Neural Network.
Die bisherigen Ergebnisse werden am Ende dieser Arbeit vorgestellt.

1 Einleitung

Oftmals wird eine Lernphase, wie beispielsweise eine Prüfungsvorbereitung, von verschiedenen Stimmungslagen begleitet. Im speziellen führen negative Emotionen wie Langeweile, Ärger oder Angst zu einem suboptimalen Lernerfolg.
Da wäre es hilfreich, wenn es einen Lern-Assistenten gäbe der die Gefühle der Lernenden/ des Lernenden erfasst, um ihr/ihm so hilfreiche Informationen zukommen lässt um diese Situation auf ein Minimum zu reduzieren.
Menschen sind in der Lage diese Emotionen durch die Gestik und Mimik anderer Menschen zu erkennen. Ebenso ist es möglich, die Gefühlslage einer anderen Person mittels ihrer Stimmlage zu detektieren. Wie sieht es jedoch mit Daten aus, die beispielsweise über ein EKG aufgenommen wurden?
Würde man jemanden eine Abbildung dieser Sensor basierten Daten vorlegen, so könnte er sehr wahrscheinlich keine Gefühle darin erkennen.

Ähnlich ist es bei Maschinen. Es ist inzwischen möglich ihnen mit Hilfe Neuronaler Netze und einer großen Menge an Bildmaterial die Erkennung von Emotionen auf Gesichtern beizubringen. Auch hier ist die Verwendung von Sensordaten ein schwieriges Unterfangen.

In diesem Forschungsprojekt werden nun zwei Architekturen von Neuronalen Netzen vorgestellt, die für diese Aufgabe als vielversprechend angesehen werden.
Nach einem Überblick zu verschiednen relatierten Arbeiten, folgt eine Einführung in die Funktionsweise von eben diesen Netzen. Anschließend werden die speziellen Netzwerke näher erläutert, sowie auf relevante Publikationen eingegangen.
Den Abschluss bilden die Ergebnisse der ersten Implementationen der betrachteten Netzarchitekturen.

2 Related Work

Die Erkennung und Klassifikation von Emotionen mit Hilfe Neuronaler Netze ist bereits Gegenstand vieler Publikationen. Dafür wurden verschiedenste Datengrundlagen herangezogen. Unter anderem auch eindimensionale Sensordaten, wie auch im Falle dieses Projektes.
In der Arbeit von Lee et al. kam beispielsweise ein Multilayer Perzeptron zum Einsatz, welches mittels EDA und HRV Daten eine Klassifikation ermöglichen sollte [LYP+06].

Weitere Arbeiten, in denen die klassische Struktur von Feed Forward Netzen verwendet wurde, sind in der Arbeitsgruppe von Malaveras et al. [MSD+98] und Fernandez et al. enstanden [FVHPÁEMB16]. Malvares verwendet für seine Analyse Daten Grundlage durch EKG. Fernandez hingegen setzt eine Detektion des Arousal Levels durch Verwendung von EEG Daten um.

Eine Publikation, die einen Einfluss auf die Wahl der Netzarchitekturen in dieser Arbeit hat, war jedoch eine Empirische Studie von Bai et al. [BKK18]. Bai erforschte dabei, welche Neuronale Netzarchitektur für die Verarbeitung von Sequenzen am vielversprechendsten erscheint.

Bei der weiteren Recherche fielen zudem noch die Werke von Greco et al. und Caho et al. auf [GVCS17][CTY+15].

Greco erforschte die Erkennung von Valence und Arousal mit Hilfe von Convolutional Neural Networks, wohingegen Chao einen Ansatz mit Reccurent Neural Networks verfolgte. Eine Kombination dieser beiden Architekturen fand in der Gruppe um Kevin Brady Verwendung [BGK+16].

3 Emotional Picture Experiment HTW

Das für dieses Forschungsprojekt verwendete Datenmaterial wurde während des Emotional Picture Experimentes an der HTW Berlin erhoben.
In diesem Versuch haben 27 Probanden anhand von emotionsstimulierenden Fotografien ihre Gefühlslage bewertet.

Während des jeweiligen Versuchszeitraums von ca. 30 Minuten sind die Teilnehmenden mit Sensoren zur Messung ihrer elektrodermalen Aktivität und Herzfrequenz (EKG) verbunden gewesen. Bei der elektrodermalen Aktivität handelt es sich um die Messung des Leitungswiderstandes der Haut.
Dieser Widerstand wird durch erhöhte oder verringerte Schweißproduktion beeinflusst. Von den Probanden wurden einmalig jeweils 96 verschiedene Bilder aus dem International Affective Picture System (IAPS) betrachtet und bewertet.

Das IAPS ist ein von Speziallisten aus dem Bereich der Psychologie entwickeltes System. Es wird verwendet, um über das enthaltene Bildmaterial, Emotionen wie beispielsweise Vergnügen (amusement) oder Zufriedenheit (contentment), auszulösen [LBC97][LB07].
Beispiele für diese Bilder sind in Abbildung 1 dargestellt. Zusätzlich beinhaltet es ein Bewertungssystem, mit dem der Betrachtende seinen Gefühlszustand beurteilt.

(a) Ein Beispiel für Vergnügen

(b) Zufriedenheit

Abbildung 1: Bilder aus dem Testdatensatz des International Affective Picture Systems.
Bildquelle: https://www.imageemotion.org/

Die Bewertungskriterien beziehen sich auf den Grad der Erregung (Arousal) und der Wertigkeit (Valence) des Gefühlten. Die Klassifizierung der Gefühlslage erfolgt durch das von Frau Prof. L. Feldmann Berett vorgestellte Valence-Arousal Circumplex [Fel95]. Abbildung 2 zeigt das besagte Diagramm.

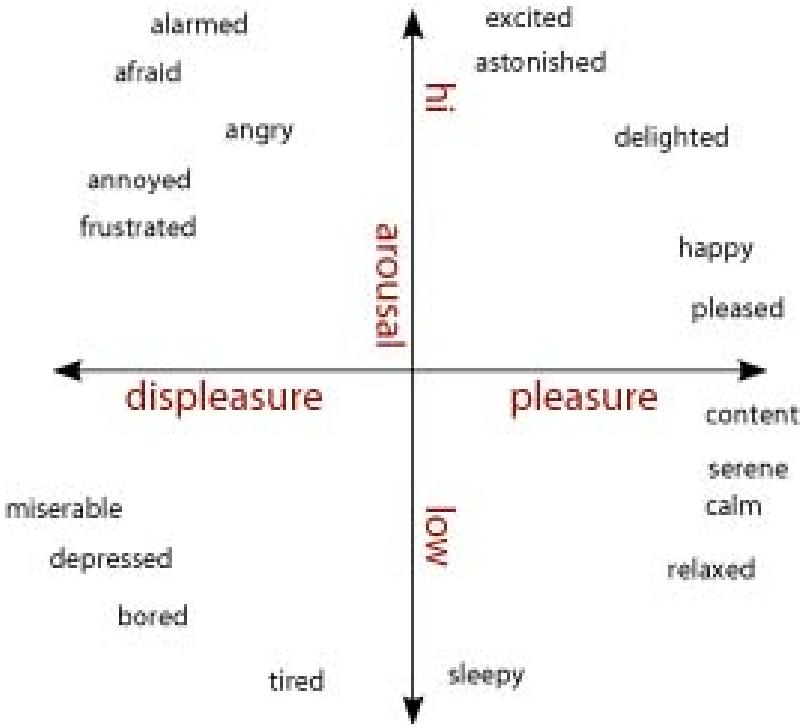
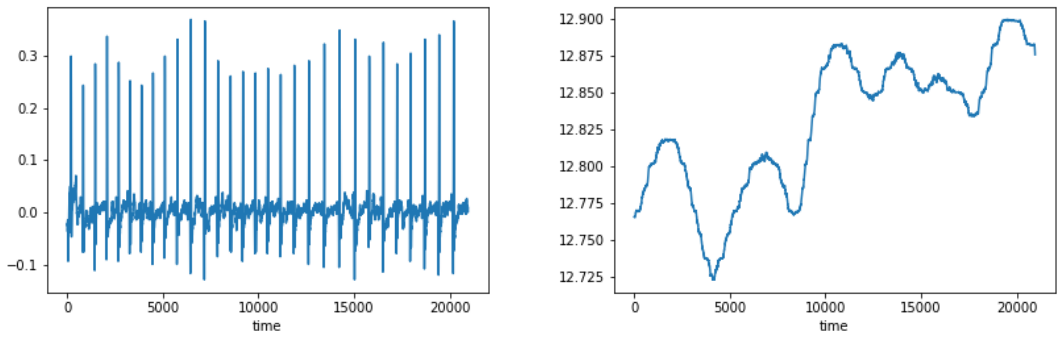


Abbildung 2: Das Valence-Arousal Circumplex nach L. Feldman Berett.
Bildquelle: *https* : *//en.wikipedia.org/wiki/Emotional*$_g$*ranularity*

Zu erkennen sind vier Bereiche in die sich Emotionen gliedern lassen. Ein hohes Arousal und eine positive Valence spiegeln beispielsweise Fröhlichkeit oder Aufregung wieder, wohingegen niedrige Erregung und negative Wertigkeit als deprimiert oder gelangweilt interpretiert werden können.

Zur Betrachtung und Bewertung eines Bildes standen den einzelnen Teilnehmenden 21 Sekunden zur Verfügung. Dieser Zeitraum unterteilt sich in drei Phasen, bei der die erste eine fünf sekündige Vorbereitungsphase ist. Ihr folgt die Betrachtungsphase, bei dem den Probanden das Bildmaterial gezeigt wird. Zum Abschluss verbleiben zehn Sekunden für die persönliche Bewertung des Arousal und Valance Levels.

Zur Veranschaulichung des Datenmaterials für einen 21 sekündigen Zeitraum, wurden die Sequenzen geplottet und in Abb. 3 dargestellt.



(a) EKG Signal

(b) EDA Signal

Abbildung 3: Aufgezeichnete Signale für den 21 sekündigen Zeitraum eines Bildes

4 Neuronale Netze

Um ein Verständnis für die grundlegende Funktionsweise von Neuronalen Netzen zu vermitteln, wird zu Beginn dieses Abschnitts das allgemeine Prinzip erläutert.

Vertiefend wird auf die in diesem Forschungsprojekt betrachteten Architekturen eingegangen. Das Kriterium welches zu der Wahl der Netztypen geführt hat, ist die Eignung zur Verarbeitung der zugrundeligenden Daten. Wie im vorherigen Kapitel bereits erwähnt, handelt es sich um ein dimensionales Datenmaterial.

Bei der Beschreibung der spezifischen Neuronalen Netz Architekturen wird darauf geachtet Beispiele anhand der gegebenen Daten zu verwenden, um einen direkten Kontext zu gewährleisten.

4.1 Grundlagen Neuronaler Netze

Neuronale Netze kommen aus dem Gebiet des Maschinellen Lernens. Sie sind speziell im Bereich der künstlichen Intelligenz angesiedelt. Als Vorlage für ihren Aufbau wurden sie dem biologischen Vorbild nachempfunden.

Die aus dem Nervensystem von Lebewesen inspirierte Form des künstlichen Äquivalents, richtet sich dabei nach der Übertragung von Informationen durch miteinander verbundene Neuronen. Wird im Vergleich die Graphendarstellung eines künstlichen Neuronalen Netzes betrachtet, so lassen sich die Knoten als künstliche Neuronen und die Kanten als deren Verbindungen verstehen (Abb. 4).

Befinden sich mehrere Neuronen untereinander, also auf einer Ebene, werden diese als Schichten oder Layer bezeichnet. Es ist auch möglich, dass sich ein einzelnes Neuron in einer Schicht befindet. Von diesen Layern gibt es im gesamten drei Typen, die Eingabe- oder Inputschicht, die Versteckten- oder auch Hiddenlayer und die Ausgabeschicht, auch als Outputlayer bezeichnet. Abbildung 4 zeigt ein einfaches Feedforward Netz. Dies ist ein Netz, in dem die Richtung ausschließlich von der

Eingabe über die Versteckten bis zur Ausgabeschicht fließt. Zu erkennen sind hier die einzelnen Schichten mit ihren Neuronen, so wie die Verbindungen zwischen den Neuronen der einzelnen Ebenen.

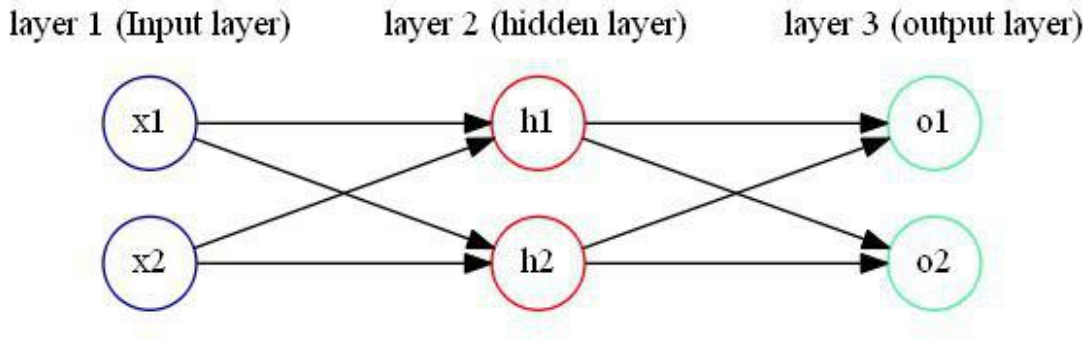


Abbildung 4: Einfaches Feedforward Netz mit Input, Hidden und Output Layer. In grün hervorgehoben sind gewichtete Verbindungen zwischen spezifischen Neuronen.

Die Funktionsweise der künstlichen Neuronen kann in vier Komponenten unterteilt werden. Das erste Element bildet die über die Verbindungen gewichtete Eingabe. Eine Ausnahme bildet die Inputlayer, da sie die Initiale Ebene darstellt. Anhand der Gewichtung wird festgelegt welchen Einfluss die vorherigen Neuronen auf das Nachfolgende haben. Im zweiten Modul wird dieser Input von der sogenannten Übertragungsfunktion verarbeitet. Das Ergebnis dieser Funktion wird auch als Netzeingabe bezeichnet. In der Regel handelt es sich um eine Aufsummierung der gewichteten Eingaben wie in der Abbildung 5 gezeigt wird.

$$net_j = \sum_{i=1}^{n} x_i w_{ij} \tag{1}$$

Abbildung 5: Übertragungsfunktion eines künstlichen Neurons. x_i entspricht der Eingabe der vorherigen Neuronen, die mit den enstprechenden Gewichtungen w_{jj} multipliziert werden.

Das dritte Glied bildet die Aktivierungsfunktion. Durch diese wird anhand der übergebenen Netzeingabe, der Output des Neurons bestimmt. Zu den Aktivierungsfunktionen zählen beispielsweise lineare, binäre und sigmoide Funktionen. Je nach Anwendungbereich wird entschieden, welche dieser Funktionen am sinnvollsten erscheinen. Das vierte und letzte Element wird als Bias oder Schwellwert bezeichnet. Es handelt sich dabei um einen konstanten Wert, der dazu dient, die Aktivierung eines Neurons zu steuern. So wird beispielsweise die erhaltene gewichtete Eingabe verstärkt falls diese positiv ist, oder abgeschwächt falls eine negative Netzeingabe vorliegt.

Mit dem bekannten Aufbau wird sich nun dem Training oder "Lernenëines Netzwerks gewidmet. Das Lernen eines Neuronalen Netzes lässt sich grob in zwei Felder unterteilen. Das supervised learning (beaufsichtigte Lernen) und das unsupervised learning (unbeaufsichtigte Lernen).
Im beaufsichtigten Bereich wird das Neuronale Netz anhand von Beispielen trainiert, wohingegen das unbeaufsichtigte Verfahren durch Ähnlichkeiten des gegebenen Datenmaterials lernt. Somit eignen sich die beaufsichtigten Algorithmen besonders für Klassifizierungsaufgaben und die unsupervised für Segmentierung (Clustering). Da im Falle dieses Projektes eine Klassifizierung von Emotionen angestrebt wird sind im Vorfeld ausschließlich Netzwerk Architekturen in Betracht gezogen worden, die einen beaufsichtigten Ansatz verfolgen. Aufgrund dessen erfolgt die Beschreibung des Lernprozesses alleinig auf der Grundlage des supervised Learning.

Die während des "Lernvorgangsïm Zentrum stehenden Komponenten sind die Gewichtungen der einzelnen Verbindungen. Diese werden zufällig initialisiert und während des iterativen Trainingsprozesses schrittweise optimiert. Dies geschieht in vier sich wiederholenden Aktionen. Begonnen wird mit dem Forwardpath, bei dem die Daten alle Schichten des Netzes, von Input bis zur Output Layer, durchlaufen. Zur Veranschaulichung dienen die grünen Kanten des Graphen in Abbildung 6. Mittels der Egebnisse aus der Ausgabeschicht kann nun die nächste Aktion, das Berechnen des Fehlers, erfolgen.

Als Beispiel dient wiederum Abb. 6. Die zwei Neuronen der Output Layer repräsentieren hier die zur Verfügung stehenden Klassen. Um den Prozess zu Verbildlichen werden diese als "Fröhlichöder "Traurig"bezeichnet.

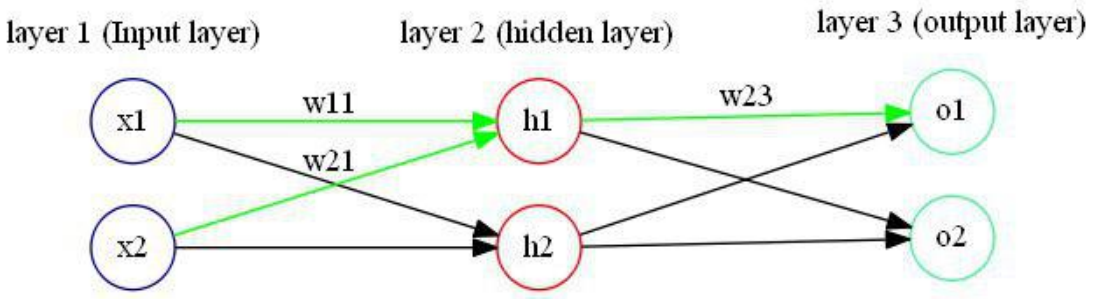


Abbildung 6: Einfaches Feedforward Netz mit Input, Hidden und Output Layer. Gewichtete Verbindungen zwischen spezifischen Neuronen sind sind in grün hervorgehoben.

Geben die Neuronen der letzten Schicht nun Werte wie etwa 0.6 und 0.4 aus, so ist das Netz der Überzeugung, dass die verarbeiteten Daten der Klasse "Fröhlichënt-sprechen. Da mit Beispielen trainiert wird, liegen Wahrheits - oder auch Ground Truth Daten vor.
Mit Hilfe dieser Wahrheitswerte kann jetzt der entstandene Fehler (engl. Loss oder Error) berechnet werden. Hierzu dienen sogenannte Loss- oder Error-Funktionen. Eine häufig verwendete Methode zur Fehlerberechnung ist zum Beispiel der Mean Squared Error (MSE), welcher in Abb. (7) aufgeführt ist.

$$MSE = \frac{1}{2}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2}$$

Abbildung 7: Der Mean Squared Error bildet die Summe aus den Quadrierten Differenzen aus Vorhersagen Y_i und Wahrheitswerten $\hat{Y}_i$.

Der Loss gibt Aufschluss darüber, wieviel Abstand zu dem Wahrheitswert besteht und durchläuft mit diesem "Wissen"das Netz nun rückwärts (siehe Abbildung 8).

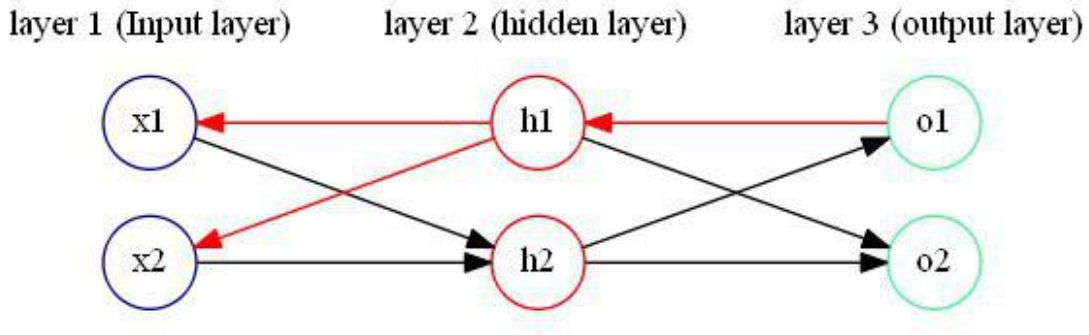


Abbildung 8: Die roten Pfeile stellen den Fehler dar, der rückwärts durch das Netz läuft und dabei die Gewichtungen optimiert.

Dieser Abschnitt wird als Backpropagation bezeichnet und dient der Ermittlung der Gradienten, in dessen Richtung die jeweiligen Gewichtungen angepasst werden sollen. Damit die Berechnung der Schrittbreite zur Änderung der Gewichte dynamisch und granular erfolgt, wird das Gradientenabstiegsverfahren oder im englischen Gradient Descent verwendet.
Da das gesamte Netz eine Aneinanderreihung von mathematischen Operationen darstellt, kann diese Methode verwendet werden, um anhand der Kettenregel die Ableitung der einzelnen Funktionen zu berechnen. Das daraus reultierende Ergebnis ist als der Gradient oder die Steigung der einzelnen Abschnitte zu interpretieren. Die Formel in Abb. 9 stellt diesen Prozess dar.

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial o_k} * \frac{\partial o_k}{\partial w_{jk}} \tag{3}$$

Abbildung 9: Ableitung anhand der Kettenregel im Gradient Descent Verfahren.

Mit den an diesem Punkt bekannten Steigungen, lassen sich im letzten Abschnitt die einzelnen Gewichte optimieren. Der neue Wert der einzelnen Wichtungen erfolgt über die Subtraktion der partiellen Ableitung, von dem aktuellen Wichtungswert. Dies kann aus Abb. 10 entnommen werden. Die in der Formel verwendete Konstante α wird als Lernrate bezeichnet und sorgt für eine Dämpfung der einzelnen Änderungen. Dies dient der Vermeidung von zu "großenSSchritten, welche dazu führen

könnten, dass das zu erzielende Minimum des Fehlers überschritten wird. Das Überschreiten des Minimums kann zur Folge haben, das der Fehler wiederum ansteigt.

$$neu\ w_{jk} = alt\ w_{jk} - \alpha * \frac{\partial E}{\partial w_{jk}} \tag{4}$$

Abbildung 10: Ableitung anhand der Kettenregel im Gradient Descent Verfahren.

Diese Operationen wiederholen sich jetzt so häufig, bis entweder die maximale Anzahl von Iterationsschritten (auch als Epochen bezeichnet) durchlaufen wurde, oder das Optimum bereits frühzeitig gefunden wurde.

4.2 Convolutional Neural Networks

Convolutional Neural Networks, abgekürzt CNN, sind die ersten speziellen Netzarchitekturen, die in diesem Forschungsprojekt betrachtet wurden.
Die von Yan LeChun [LBB+98] ursprünglich entwickelte Anwendung eines solchen Netzwerks liegt im zweidimensonalen Raum. Speziell diente sie zur Erkennung von handgeschriebenen Zahlen und wurde vor allem im Bankensektor zur schnelleren Verarbeitung von Schecks eingesetzt [LBB+98].
Da die Wurzeln dieser Architektur im zweidimensionalen Raum liegen und deren häufigsten Anwendungsfälle im Bereich der Bildklassifikation zu finden sind, soll die Funktionsweise anhand einer klassischen Architektur illustriert werden 11.

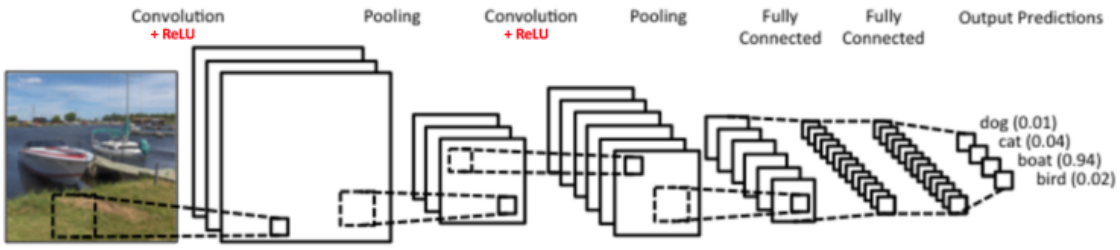


Abbildung 11: Aufbau eines zweidimensionalen Convolutional Neural Network. Zu erkennen sind die wiederholt auftretenden Convolutional- und Pooling Layer. Zum Ende wird eine vollvermaschte Schicht angehängt, die in einer Klassifizierung endet. Bildquelle:http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

Als Eingabe dient ein Bild, welches innerhalb des Netzes als Matrix verwendet wird. Diese Matrix durchläuft nun das gesamte Netzwerk und beginnt mit der Convoultional Layer.
Ihren Namen hat die Schicht von der hier verwendeten mathematischen Operation, die im englischen als Convolution und im deutschen als Faltung bezeichnet wird. Das Ziel dieser Funktion ist es, anhand von Filtermasken bestimmte Features in diesem Bild hervorzuheben.
Diese Masken oder auch Kernel sind ebenfalls Matrizen, die zur Bestimmung der Features, schrittweise von links nach rechts, über die übergebene Matrix geschoben

werden. In der Regel wird eine Schrittweite von einem Pixel oder einem Matrizenelement gewählt. Erreicht der Kernel den rechten Rand der Matrix, wird er um eine Zeile nach unten versetzt und wiederholt den Prozess bis die gesamte Struktur abgearbeitet ist.
Während dieser Prozedur werden die Werte aus der Filtermaske mit denen des Inputs elementweise multipliziert und die Ergebnisse der einzelnen Multiplikationen aufsummiert. Abbildung 12 stellt dieses Verfahren bildlich dar.

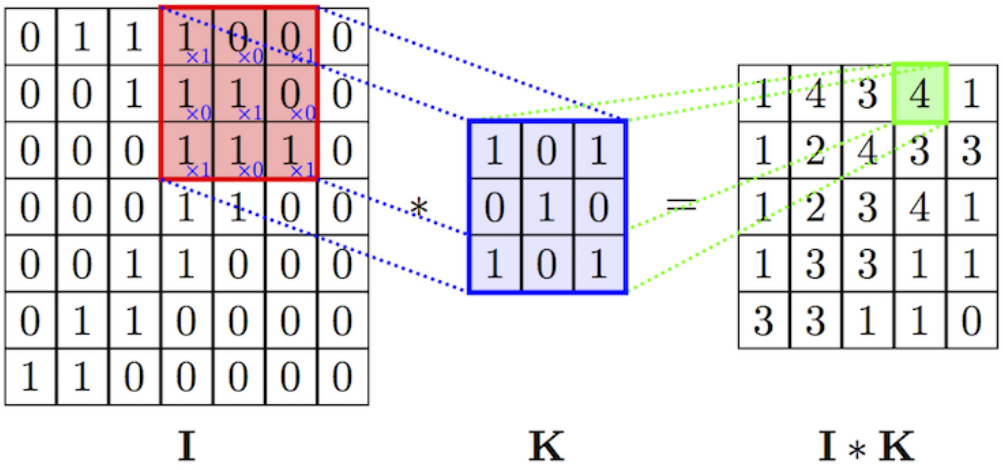


Abbildung 12: Illustation der Faltungsoperation. Ïst das Eingabebild, "K"repräsentiert die Filtermaske und $I * K$ das Resultat.
Bildquelle: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/ und

Das Resultat wird nun in eine Ergebnismatrix geschrieben. Die Zelle in die der berechnete Zahlenwert geschrieben wird, hängt dabei von dem mittleren Eintrag des Kernels ab.
Während des Trainigs sind es die Werte der Filtermasken, die optimiert werden. Somit können sie als die Gewichtungen des Netzwerks interpretiert werden.
Nachdem die Convolution abgeschlossen ist, fließt die resultierende Matrize durch die als ReLu bezichnete Aktivierungsfunktion in die Pooling Ebene. Die ReLu oder "Rectified Linear Unit"wird dazu verwendet die negativen Werte innerhalb des Ergebnisses zu neutralisieren [HSM+00] 13.

$$f(x) = max(0, x) \tag{5}$$

Abbildung 13: "Rectified Linear Unit"Funktion

Die in der Pooling Schicht verwendete Funktion wird auch als Max Pooling bezeichnet. Wie auch in der Convolution Layer kommen Filtermasken zum Einsatz. Jedoch unterscheidet sich ihre Anwendung erheblich. Zum Einen ist die Schrittbreite der Maske in der Regel so groß wie die Filtermatrix. So wandert beispielsweise eine $2x2$ Matrize zwei Schritte nach rechts und eine $3x3$ Maske drei Schritte. Zum Anderen wird von dem Filter lediglich der höchste Wert in seinem Betrachtungsraum in das Resultat dieser Ebene geschrieben. Dies hat zudem zur Folge, dass in den

Filtermasken der Poolingschicht keine Gewichte vorliegen, die während des Trainings optimiert werden können. Anhand der Abbildung 14 wird verdeutlicht welche Aufgabe die Filter dieser Ebene übernehmen.

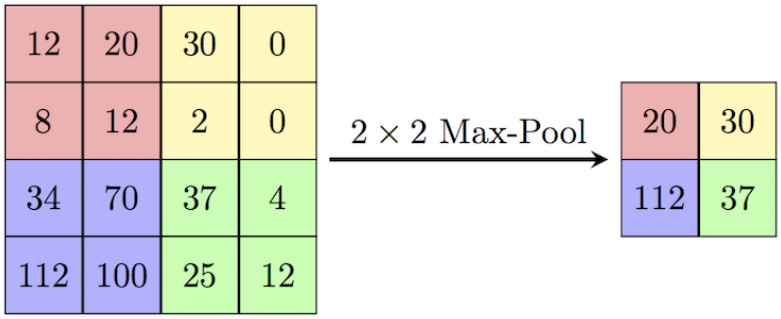


Abbildung 14: Max Pooling. Links ist die Eingabe und rechts das Resultat. Verwendet wurde für diese Filterung ein $2x2$ großer Kernel.
Bildquelle: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/ und

Nachdem das anfängliche Bild jede Convolutional und Pooling Layer durchlaufen hat, wird die Matrix zu einem Vektor geglättet und in eine Fully Connected Layer überführt. Der Aufbau und die Funktionsweise dieser Schicht sind identisch zu der im Abschnitt 6 beschriebenen Netzstruktur.
Um eine Vorstellung von den gelernten Features zu vermitteln, wird ein Beispiel aus der Gesichtserkennung in Abb. 15 dargestellt.
Hier zu sehen ist beispielsweise, dass in der ersten Layer zunächst eine Kantenerkennung vollzogen wird. In den Masken der zweiten Schicht sind Merkmale, wie Nasen, Augen und Münder zu erkennen. In der dritten dargestellten Ebene sind nun die gesamten Gesichter zu erkennen.

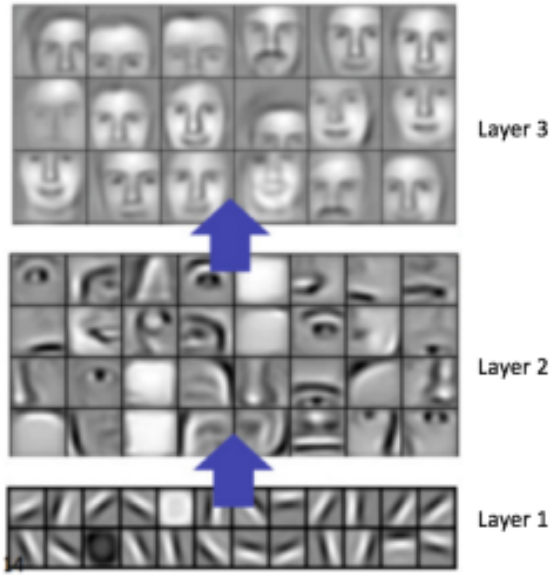


Abbildung 15: Visualisierung der durch die Filtermasken gelernten Features.
Bildquelle: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/ und[LGRN09]

Der größte Vorteil dieser Architektur ist die hohe Accuracy [LBH15]. Die Accuracy ist ein Messwert, der eine Aussage darüber gibt, wie genau das Netz generalisiert und somit auch klassifiziert. Eine Accuracy von 90% besagt also, dass 90% der klassifizierten Daten korrekt waren.
Zu den Nachteilen dieser Netze gehört der hohe Rechenaufwand [LBH15]. So kann es beispielsweise sein, dass bei einer komplexen Klassifizierungsaufgabe die GPUs einer leistungsstarken Grafikkarte benötigt werden, um die Dauer des Trainings zu minimieren.
Ein weiterer Nachteil ist, dass in den meisten Fällen ein sehr großes Datenset benötigt wird, um eine Generalisierung zu ermöglichen [LBH15].

Im Bereich der Emotionserkennung wurden diese Netze bereits eingesetzt. In der Arbeit von Brady et al. [BGK+16] wurden etwa Video daten verwendet um die Emotionen der gefilmten Person zu bestimmen. Die Accuracy Werte für die Klassifizierung anhand des Bildmaterials ist der folgenden Tabelle zu entnehmen 1.

	Valence	**Arrousal**
Video	47%	48%
Audio	45%	79%
EDA	10%	7%
ECG	15%	28%

Tabelle 1: Ergebnisübersicht aus dem Paper von Brady et al. [BGK+16]

In der Tabelle ist ebensfalls erkennbar, dass neben dem zweidimensionalen Bildmaterial auch eindimensionale Zeitreihen, wie sie in diesem Projekt vorliegen, verwendet wurden.

4.2.1 1D Convolution

In diesem Forschungsprojekt werden Signale genutzt, die sich als Zeitreihen verstehen. Daher ist ein zweidimensionales CNN nicht verwendbar.
Es wäre prinzipiell möglich die geplotteten Sequenzen von EDA und ECG mit einem zweidimensionalen CNN zu verarbeiten. Dies würde jedoch bedeuten, dass die Zeitreihen erst aufgenommen und geplotted werden müssten, bevor eine Klassifizierung erfolgen kann 16.

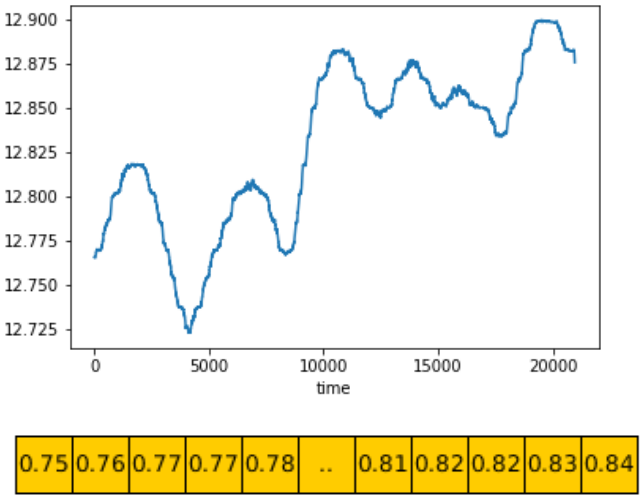


Abbildung 16: Darstellung eines EDA-Signals als Plott und als 1D Zahlenarray.

Eine Alternative ist das eindimensionale Convolutional Neural Network. Es funktioniert im Prinzip wie sein 2D äquivalent mit ein paar kleinen Unterschieden. Aus Abb. 17 ist ersichtlich, dass sich bei der Struktur kein direkter Unterschied erkennen lässt. Doch ist das Inputsignal kein Bild sondern ein Array 16. Die weiteren Feinheiten finden sich in den Funktionalitäten der Schichten.

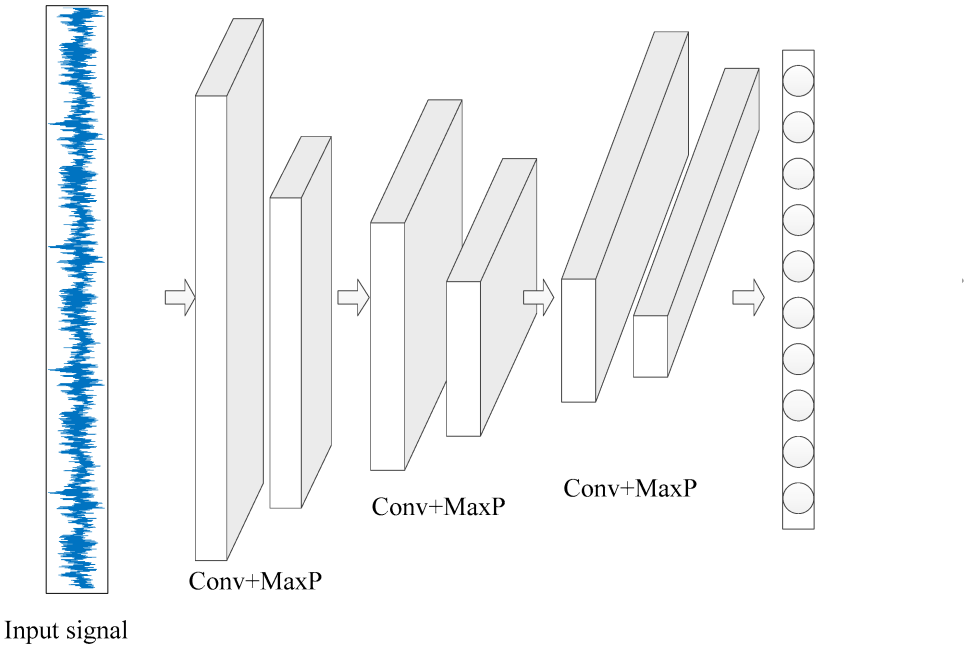


Abbildung 17: Aufbau eines eindimensionalen Convolutional Neural Network. Bildquelle: [CKW+19]

Die in der Convolution Layer verwendeten Filter sind nun ebensfalls eindimensionale Arrays. In Abbildung 18 ist zu sehen, dass wie auch in der klassichen CNN Architektur der Kernel über das Array wandert und das Resultat in einer Ergebnisliste

speichert. Die initialen Werte der Masken werden ebenfalls zufällig initalisiert und durch das Training optimiert.

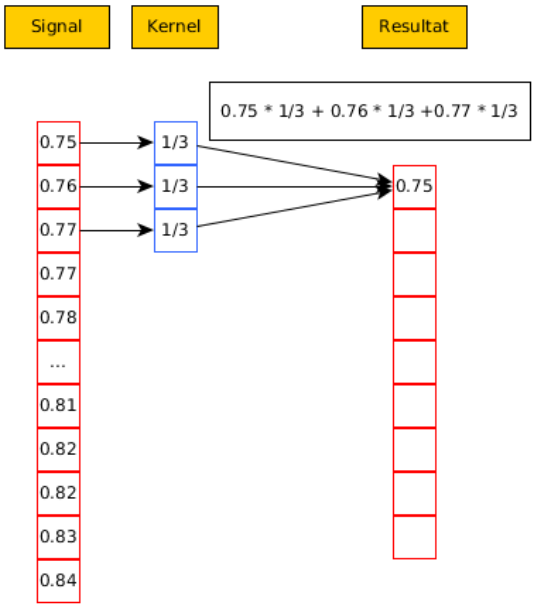


Abbildung 18: Die Konvolutionsschicht in einem CNN.

Der einzige Unterschied zu der Poolingschicht liegt in der Form der Kernel. Sie sind in dieser Architektur ebenfalls eindimensional (siehe Abb. 19. Die Funktionsweise ändert sich jedoch nicht.

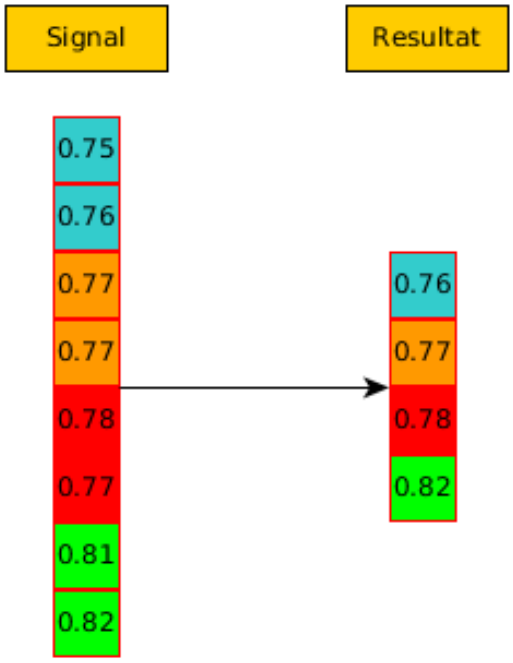


Abbildung 19: Funktionsweise der Max-Poolingschicht in einem 1D Convolutional Neural Network.

Diese eindimensionalen Netze sind laut der empirischen Studie von Bai et al. [BKK18] für Aufgaben im Bereich des Sequenzmodelings die beste Wahl.
Neben dieser Studie gibt es allerdings noch weitere Arbeiten die ein 1D CNN verwenden, um Zeitreihen zu Klassifizieren.
So haben beispielsweise Keren et. al Implementationen dieser Architektur genutzt, um eine Bestimmung des Arousal- und Valence-Levels zu ermöglichen [KKM+17]. Die in diesem Paper erzielten Resultate sind in Tabelle 2 dargestellt.

	Valence	**Arrousal**
Development	48%	46%
Test	41%	43%

Tabelle 2: Ergebnisübersicht aus dem Paper von Karen et al.[KKM+17]
Das Development steht in diesem Fall für die Accuracy während des Trainings. Der Test hingegen steht für die Resultate der abschließenden Testphase.

Ein weiterer Ansatz zur Klassifikation von Arrousal und Valence mit Hilfe von CNN wurde 2017 von Greco et al. [GVCS17] vorgestellt. Der Zweck dieser Arbeit war es mit verschiedenen Features, wie beispielsweise der Raum unterhalb einer EDA-Kurve, das Aurousal und Valence Level einer Person zu klassifizieren. Die erzielten Ergebnisse sind in Tabelle 3 und 4 dargestellt.

	Positiv valence	**Negativ valence**
Positiv valence	84%	16%
Negativ valence	16%	84%

Tabelle 3: Konfusionsmatrix aus dem Paper von Greco et al. [GVCS17].
Die Werte geben an, zu wieviel Prozent die gegebenen Klassen (negativ und positive Valence) korrekt bestimmt wurden.

	AR1	**AR2**	**AR3**
AR1	72%	8%	8%
AR2	12%	88%	12%
AR3	16%	4%	80%

Tabelle 4: Konfusionsmatrix aus der Arbeit von Greco et al. [GVCS17].
Auch hier ist die prozentuale korrekte Klassifizierung dargestellt. Greco et al. unterteilte das Arousal in drei Level

Zudem kamen sie in der Arbeit von Strober et al. zum Einsatz [SSOG15]. Die Arbeitsgruppe um Strober analysierte mit Hilfe einer Kombination aus Autoencoder und Convolutional Neural Networks elektroenzephalografisches Datenmaterial. Das Ziel ist es gewesen, neue Features innerhalb dieses Signals ausfindig zu machen.

4.3 Recurrent neural network

Recurrent Neural Networks oder kurz RNN wurden designed um in Sequenzen und/oder Zeitreihen gewisse Muster oder Feature zu entdecken [MJ01]. Aufgrund dessen, dass es sich bei den erhobenen Daten um Zeitreihen handelt, wirkt ein RNN vielversprechend.
RNN's sind so konzipiert, dass in einer Sequenz die jeweilig vorherigen Zeitpunkte einen Einfluss auf das Ergebnis haben. Betrachtet man Abbildung 20, so sind hier die verschiedenen Komponenten des Netzwerks zu erkennen.

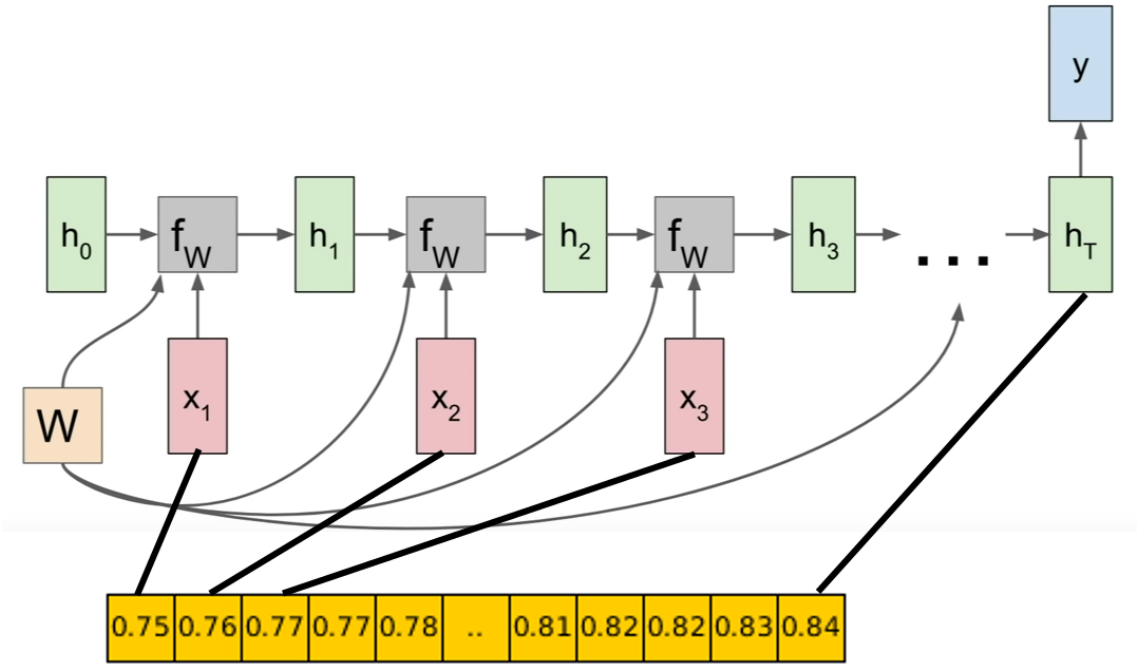


Abbildung 20: Darstellung eines Recurrent Neural Networks. Die Eingabe ist als Array dargestellt und verweist auf die jeweiligen Punkte, zu denen sie in das Netz eingebracht werden.

Die mit x_i bezeichneten Werte sind die einzelnen Zeitschritte, die in das Netz eingebracht werden. Zur Verdeutlichung dient die in gelb eingefügte Zeitreihe, die beispielsweise als ein EDA Signal verstanden werden kann.
Ebenfalls zu erkennen sind die mit h_i gekennzeichneten Hiddenstates. Der Wert den dieser Bereich annimmt wird in der Funktion f_w ermittelt und gibt Aufschluss über den Zustand des Netzes zum Zeitpunkt x_i. Initialisiert wird der Hiddenstate h_0 meistens mit einem Wert von null.
Somit wird gewährleistet, das die Resultate der folgenden Zellen diesen Status in die Brechnungen mit einbeziehen.
Die Aktivierungsfunktion f_w ist in der klassichen RNN Architektur eine Tangens Hyperbolicus Funktion 21. Sie übernimmt die mit den Gewichten multiplizierten Hiddenstates und den gewichteten aktuellen Zeitschritt als Eingabe.

$$h_t = tanh(w_{hh}ht - 1, w_{xh}xt) \tag{6}$$

Abbildung 21: Tangens Hyperbolicus. Wird verwendet um den hiddenstate in einem Klassichen RNN zu berechnen.

Einen Unterschied zu den klassischen Feed Forward Netzen und den CNN's ist, dass in einem RNN eine globale Gewichtung vorliegt. Was bedeutet, dass zu jedem Zeitschritt auf die gleiche Gewichtungsmatrix zugegriffen wird.

Vorteile dieser Architektur sind, dass sie für die Analyse von nicht linearen Zeitreihen ausgerichtet sind. [MJ01]. Nachteile dieser Architektur sind, wie bei einem großteil der Neuronalen Netze, dass sie eine große Menge an Daten benötigen, um ein solides Ergebnis erzielen zu können.
Die beiden größten Nachteile eines klassischen Recurrent Neural Networks sind jedoch das Problem des Exploding- und Vanishing Gradients. Unter Exploding Gradient ist zu verstehen, dass während der Backpropagation zu große Gradienten ermittelt werden können, sodass ein Lernerfolg des Netzes nicht mehr möglich ist [PMB12].
Das Vanisching Gradient Problem ist das Pendant zu dem beschriebenen Exploding Gradient. In diesem Fall nähert sich der Gradient null an und sorgt insofern dafü, dass ein Trainingserfolg ausbleibt[PMB12].

4.3.1 LSTM

Die von Hochreiter et al. im Jahr 1997 vorgestellten Long Short Term Memory (LSTM) Zellen [HS97] sind entwickelt worden, um die Problematiken des Vanishing und Exploding Gradient zu beheben. Nachfolgend werden die Unterschiede der beiden Architekturen näher erläutert.
In Abbildung 22 ist die klassische Implementation einer Recurrenten Zelle dargestellt. Zu erkennen sind hier die aus vorherigen Kapitel bekannten Elemente. Der als x_t beschriebene Input des Zeitschritts, der Tangens Hyperbolicus als Aktivierungsfunktion und der als h_t gekennzeichnete Hiddenstate

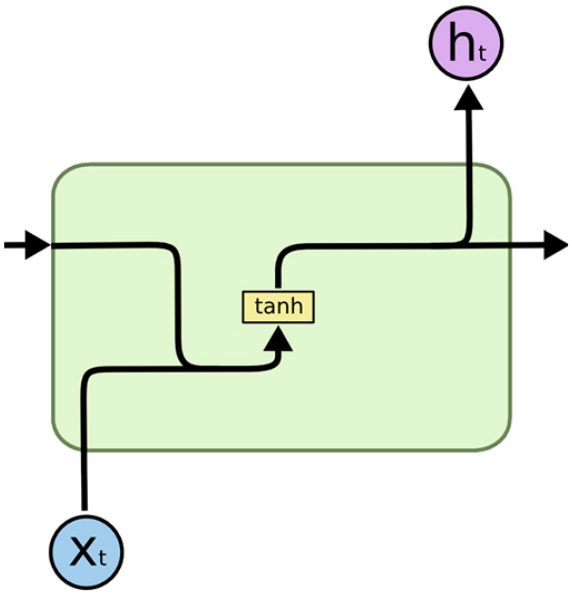


Abbildung 22: Klassiche RNN-Zelle mit dem Tangens Hyperbolicus als Aktivierungsfunktion.
Bildquelle:http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Wird nun im Vergleich die Abbildung 23 betrachtet, so sind die Unterschiede sehr offensichtlich.
Das LSTM verfügt neben der Tangens Hyperbolicus Funktion über drei weitere Bauteile. Diese Module werden als Gates (Tore) bezeichnet und haben jeweils eine bestimmte Aufgabe zu erfüllen.
Ein weiteres wichtiges Merkmal ist, dass jede Zelle neben dem Hidden State auch einen Cell State besitzt.

Der Zustand der Zelle ist die Hauptkomponente des LSTM [HS97]. Er kann als ein Förderband betrachtet werden, das alle Informationen beinhaltet und diese durch die Zellen führt. Dabei erhält dieser Status nur geringfügige Anpassungen.
Der Cell State wird ausschließlich durch die Interaktion mit den Gates beeinflusst und hat eine signifikante Wirkung auf den späteren Hidden State.
Zur Beschreibung der Funktionalitäten werden die einzelnen mit σ gekennzeichnten Methoden der Reihenfolge nach erläutert.

Das erste Gate wird als das Forget Gate bezeichnet und trägt die Verantwortung dafür, ob die Informationen aus der vorherigen Zelle "vergessenöder "behalten"werden sollen. Konkret erfolgt hier eine Abschätzung, ob der Einfluss der vorherigen Zelle relevant ist oder nicht. Diese Evaluation erfolgt über die Sigmoid Funktion die diesem Gate zugrunde liegt. Ist das Resultat der Berechnung 0, wird die Information zum Zeitschritt $t-1$ aus dem Cell State entfernt. Ist er auf der anderen Seite 1, wird durch das Gate signalisiert diese Informationen zu behalten.

Das zweite Gate, welches als Input Gate bezeichnet wird, generiert mit Hilfe des dritten Gates, welches keinen speziellen Namen trägt, den Kandidaten für den neuen Cell State.

Im Input Gate wird wiederum eine Sigmoide Funktion verwendet, um die relevanten Daten aus der Eingabe zu filtern. Mit diesen Informationen wird das Resultat des dritten Gates, welches der Funktionsweise der klassichen Aktivierung entspricht, multipliziert und generiert somit einen Kandidaten für den neuen Cell State. Ist im Forget Gate die Entscheidung getroffen worden, dass der vorherige Zeitschritt vergessen werden soll, so wird dieser mit dem neuen Kandidaten ersetzt.
Abschließend erfolgt die Operation des Output Gates. Dieses Tor verwendet, wie das Input und Forget Gate, eine Sigmoid Funktion, um die für die Ausgabe interessanten Daten zu optimieren. Die Optimierung erfolgt dabei durch die Multiplikation der Output Gate Ausgabe mit dem aktuellen Cell State.

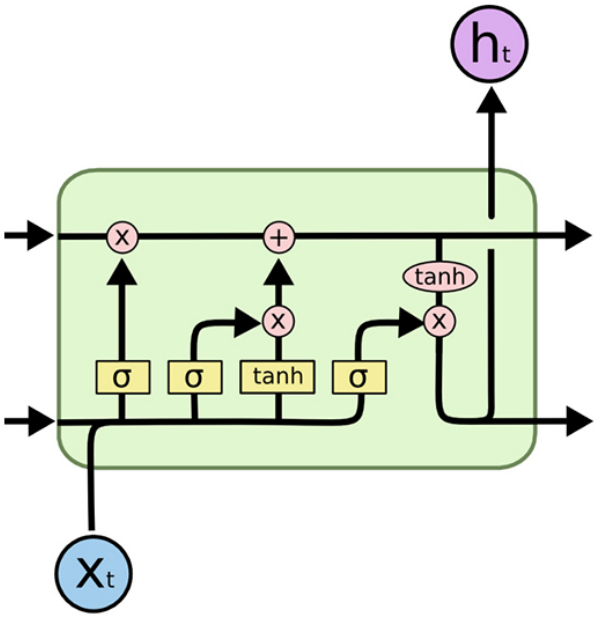


Abbildung 23: Eine Zelle in einem Long Short Term Memory Netzwerk.
Bildquelle:http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Der Ansatz zur Klassifikation mit diesem Netzwerktypen ist in der Arbeit von Chao et al. zu finden [CTY+15]. Wie auch in diesem Forschungsprojekt verwendete die Arbeitsgruppe um Chao EDA und ECG Signale, um ein Arousal und Valence Level zu bestimmen.

5 Auswertung

Dieses Kapitel bezieht sich auf die Ergebnisse aus den während des Projektes enstandenen ersten Netzwerkimplementationen. Das erste Netz welches mit Hilfe des PyTorch Frameworks in der Programmierschprache Python geschrieben wurde, war ein eindimensionales Convolutional Neural Network.
Das mit den Daten aus dem HTW Emotional Picture Experiment trainierte Netz lernte zwischen fünf möglichen Klassen zu unterscheiden.
Als Label für die Trainingsphase wurden die self ratings der einzelnen Probanden genutzt. Abbildung 24 zeigt auf der linken Seite die Accuracy-Werte während des Trainings und auf der rechten Seite die Ergebnisse der Testphase.

Train set report:

	precision	recall	f1-score	support
0	0.49	0.61	0.54	5731
1	0.44	0.36	0.40	2082
2	0.53	0.25	0.34	2771
3	0.47	0.33	0.39	2884
4	0.48	0.59	0.53	5615
micro avg	0.48	0.48	0.48	19083
macro avg	0.48	0.43	0.44	19083
weighted avg	0.48	0.48	0.47	19083

(a) Training Report

Test set report:

	precision	recall	f1-score	support
0	0.42	0.55	0.48	1434
1	0.33	0.26	0.29	520
2	0.49	0.23	0.31	663
3	0.34	0.24	0.28	700
4	0.44	0.51	0.47	1454
micro avg	0.41	0.41	0.41	4771
macro avg	0.40	0.36	0.36	4771
weighted avg	0.41	0.41	0.40	4771

(b) Test Report

Abbildung 24: Resultate der ersten Implementation eines Convolutional Neural Networks

Während der Trainingsphase konnte eine Accuracy von durchschnittlich 48% erzielt werden. Die darauf folgende Testphase gibt Aufschluss darüber, wie genau das trainierte Netz generalisiert.
Mit für das Netz unbekannten Daten erzielte es eine Accuracy von durchscnittlich 41%.

6 Ausblick

Der zweite Teil des Forschungsprojekts wird sich zum Einen mit der Optimierung der bisherigen Ansätze und zum Anderen mit weiteren Analysen zu möglichen Netzwerkarchitekturen auseinandersetzen.
Zu den Optimierungsmöglichkeiten zählen:

- Anpassung der Hyperparameter wie beispielsweise der Lernrate
- Hinzufügen von weiteren Schichten in den bereits vorhandenen Implementationen
- Untersuchung von weiteren Methoden, um die gegebenen Signale vorzuverarbeiten
- Verwendung eines größeren Datensatzes
- Untersuchen, welche Klassifikation am lukrativsten erscheint.
  - Zwei-Klassenproblem:
    - hohes/niedriges Arousal
    - positive/negative Valence
  - Vier-Klassenproblem:
    - hohes Arousal mit positiver Valence
    - hohes Arousal mit negativer Valence,
    - niedriges Arousal mit positive Valence
    - niedriges Arousal mit negativer Valence
- Untersuchung der einzelnen Phasen in der aufgenommenen Zeitreihe

Eine weitere zu betrachtende Architektur könnte beispielsweise ein sogenannter Variational Autoencoder in combination mit einem CNN und RNN sein

Literatur

[BFHIY17] Volodimir Brovkov, Albrecht Fortenbacher, Johann Habakuk Israel, and Haeseon Yun. A mobile sensor device for learning support: Design considerations. In *V Ukrainian-German conference „Informatics. Culture. Technology "*, pages 30–31, 2017.

[BGK+16] Kevin Brady, Youngjune Gwon, Pooya Khorrami, Elizabeth Godoy, William Campbell, Charlie Dagli, and Thomas S Huang. Multi-modal audio, video and physiological sensor learning for continuous emotion prediction. In *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge*, pages 97–104. ACM, 2016.

[BKK18] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[CKW+19] Xi Chen, Fotis Kopsaftopoulos, Qi Wu, He Ren, and Fu-Kuo Chang. A self-adaptive 1d convolutional neural network for flight-state identification. *Sensors*, 19(2):275, 2019.

[CTY+15] Linlin Chao, Jianhua Tao, Minghao Yang, Ya Li, and Zhengqi Wen. Long short term memory recurrent neural network based multimodal dimensional emotion recognition. In *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, pages 65–72. ACM, 2015.

[Fel95] Lisa A Feldman. Valence focus and arousal focus: Individual differences in the structure of affective experience. *Journal of personality and social psychology*, 69(1):153, 1995.

[FVHPÁEMB16] Isaac Fernández-Varela, Elena Hernández Pereira, Diego Álvarez Estévez, and Vicente Moret Bonillo. Automatic detection of eeg arousals. In *24th European Symposium on Artificial Neural Networks Bruges, Belgium, April 27-28-29*, volume 24. ESANN, 2016.

[GVCS17] Alberto Greco, Gaetano Valenza, Luca Citi, and Enzo Pasquale Scilingo. Arousal and valence recognition of affective sounds based on electrodermal activity. *IEEE Sensors Journal*, 17(3):716–725, 2017.

[HGSW04] Andreas Haag, Silke Goronzy, Peter Schaich, and Jason Williams. Emotion recognition using bio-sensors: First steps towards an automatic system. In *Tutorial and research workshop on a ective dialogue systems*, pages 36–48. Springer, 2004.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HSM+00] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947, 2000.

[KBK04] Kyung Hwan Kim, Seok Won Bang, and Sang Ryong Kim. Emotion recognition system using short-term monitoring of physiological signals. *Medical and biological engineering and computing*, 42(3):419–427, 2004.

[KKM+17] Gil Keren, Tobias Kirschstein, Erik Marchi, Fabien Ringeval, and Bjorn Schuller. End-to-end learning for dimensional emotion recognition from physiological signals. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 985–990. IEEE, 2017.

[LB07] P Lang and Margaret M Bradley. The international affective picture system (iaps) in the study of emotion and attention. *Handbook of emotion elicitation and assessment*, 29, 2007.

[LBB+98] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBC97] Peter J Lang, Margaret M Bradley, and Bruce N Cuthbert. International affective picture system (iaps): Technical manual and affective ratings. *NIMH Center for the Study of Emotion and Attention*, pages 39–58, 1997.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[LGRN09] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, pages 609–616. ACM, 2009.

[LYP+06] ChungK Lee, SK Yoo, YoonJ Park, NamHyun Kim, KeeSam Jeong, and ByungChae Lee. Using neural network to recognize human emotions from heart rate variability and skin resistance. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 5523–5525. IEEE, 2006.

[MJ01] Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5, 2001.

[MSD+98] Nicos Maglaveras, Telemachos Stamkopoulos, Konstantinos Diamantaras, Costas Pappas, and Michael Strintzis. Ecg pattern recognition and classification using non-linear transformations and neural networks: A review. *International journal of medical informatics*, 52(1-3):191–208, 1998.

[PMB12] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *CoRR, abs/1211.5063*, 2, 2012.

[SSOG15] Sebastian Stober, Avital Sternin, Adrian M Owen, and Jessica A Grahn. Deep feature learning for eeg recordings. *arXiv preprint arXiv:1511.04306*, 2015.

[YIF+17] Haeseon Yun, Johann Habakuk Israel, Albrecht Fortenbacher, Helena Rott, and Delia Metzler. User-centric approach to the design of a mobile learning companion. 2017.