Inhaltsbasierte Musikempfehlung mit Convolutional Neuronalen Netzwerken

Weidhas Philipp
Matr.nr: 123456
philipp.weidhas@st.oth-regensburg.de

Wildgruber Markus
Matr.nr: 123456
markus.wildgruber@stud.oth-regensburg.de

Zusammenfassung

Hier kommt die Zusammenfassung...

1. Einleitung

Im ersten Halbjahr des Jahres 2017 wurden 62% der Einnahmen der amerikanischen Musikindustrie durch Streaming Plattformen[1] erzielt. Im Vergleich zu Vorjahr erhöhten sich dadurch die Einnahmen um 48% auf 2.5$ Milliarde [1]. Dieser Erfolg basiert nicht nur auf einer guten Verfügbarkeit der Lieder und einem günstigen Preis sondern auch auf automatischen Musikempfehlungsdiensten, welche dem Nutzer ein angenehmeres Konsumverhalten ermöglichen. Obwohl Empfehlungsdienste in den letzten Jahren viel erforscht wurden, ist das Problem der Musikempfehlung sehr komplex. Neben einer großen Anzahl an verschiedenen Stile und Genres, beeinflussen sowohl soziales- und geographisches Umfeld, sowie der aktuelle Gemütszustand die Vorliebe eines Hörers. Diese Aspekte müssen in einem Musikvorschlag berücksichtigt werden. [2]
Neben der Empfehlung bestimmter Lieder sollen Empfehlungssysteme zusätzlich noch das Cold-Start Problem sowohl bei einem Benutzer[2] als auch bei einem Lied[3] überwinden. Das Cold-Start Problem besteht darin, dass noch keine Bewertungen für ein Lied vorliegen, wodurch es auch nicht vorgeschlagen werden kann. Dasselbe Problem gibt es bei einem neuen Benutzer: diesem kann kein guter Vorschlag gemacht werden, da es an Information mangelt welche Art von Musik ihm gefällt. [3]
Mit Hilfe von Convolutional Neuronalen Netzwerken (CNN) können noch nicht alle Probleme überwunden werden, aber sowohl das NSP als auch die Empfehlung werden durch die Verwendung eines CNN als Empfehlungssystem verbessert.

Der weitere Verlauf der wissenschaftlichen Arbeit ist wie folgt organisiert. Im 2. Abschnitt werden die Grundlagen der Musikempfehlung vorgestellt. Im 3. Kapitel wir der Aufbau von Neuronale Netze, soweit zwei Neuronale Netze für inhaltsbasierte Musikempfehlung beschrieben. Im letzten Abschnitt schließt diese Arbeit mit einem Vergleich der Ergebnisse der beiden Modelle. TODO

2. Grundlagen der Musikempfehlung

In dem Gebiet des Musik Information Retrieval (MIR) gibt es vier Kategorien [4] die einen Einfluss auf die Wahrnehmung von ähnlicher Musik haben.
Musikmerkmale sind Eigenschaften, welche aus dem Audiosignal eines Liedes extrahiert werden. Dazu zählen Aspekte wie der Rhythmus,

[1] wie Spotify, Apple Music, Pandora etc.
[2] New User Problem (NUP)
[3] New Song Problem (NSP)

die Melodie, die Harmonie oder die Stimmung eines Stücks.
Als *Musikkontext* versteht man alle Aspekte, die nicht aus dem Audiosignal abgeleitet werden, sondern Informationen die über ein Musikstück bekannt sind. Beispielsweise Metadaten wie der Titel eines Lieds, das Genre, Name des Künstlers oder das Erscheinungsjahr.
Die *Benutzereigenschaften* beziehen sie auf Persönlichkeitsmerkmale, wie Geschmack, musikalisches Wissen und Erfahrung oder den demographischen Hintergrund.
Im Unterschied dazu steht der *Benutzerkontext*, der sich auf die aktuelle Situation des Hörers bezieht. Dabei wird er durch seine Umgebung, seiner Stimmung oder der aktuellen Aktivität beeinflusst. [5]

Es gibt verschiedene Methoden, die in Musikempfehlungssystemen verwendet werden: kollaboratives -, merkmalsbasiertes -, kontextbasiertes Filtern und die hybride Methode. Diese werden genutzt, um Informationen aus den genannten Eigenschaften zu gewinnen und diese für Empfehlungen an den Nutzer zu verarbeiten. [6]

2.1 Matrix Factorization zu Musikbewertung

TODO glaube ich brauchen wir nicht, sollte bekannt sein TODO

2.2 Mel Frequency Cepstral Coefficients

Die Mel Frequency Cepstral Koeffizenten werden zur Analyse von Musikstücken verwendet und um Metadaten zuordnen zu können. Durch dieses Verfahren können die Tonhöhen, getrennt von der Sprache betrachtet werden. TODO genauer erklären? Todo

2.3 Bag of Words

BOW stellt ein klassisches Modell in der MIR da. Es stammt aus dem Feld der Textanalyse und wird dort Beispielsweise verwendet, um Dokumente automatisiert klassifizieren zu können. In diesem Modell wird ein Text als ansammlung von Wörtern gesehen. Diese Wörter werden gezählt und aufsummiert wie häufig das selbe Wort in einem Text auftritt. Über diese Häufung von Wörtern kann eine Klassifizierung dieses Textes erfolgen. TODO Zitat TODO In MIR wird dieses Modell in abgewandelter Form ebenfalls verwendet. Musikstücke werden mit Audiofeatures beschrieben. Abhängig davon wie häufig ein bestimmter Feature auf ein Lied zutrifft wird dies summiert.

2.4 Kollaborativer Filter

Kollaboratives Filtern prognostiziert Vorlieben eines Hörers, indem es aus unterschiedlichen Benutzer-Lied Verhältnissen lernt. Es basiert auf der Annahme, dass Verhalten und Bewertungen andere Nutzer auf eine vernünftige Vorhersage für den aktiven Benutzer schließen lassen [3]. Durch explizite[4] und implizite[5] Rückmeldung eines Hörers an das Empfehlungssystem empfiehlt dieses neue Lieder, indem es Gemeinsamkeiten auf Basis seiner Bewertungen mit dem Nutzungsverhalten anderer Anwender der gleichen Plattform vergleicht [7].
In der praktischen Umsetzung bedeutet dies: hört ein Anwender ein bestimmtes Musikstück. Dann werden ihm, von der Empfehlungsplattform, Lieder vorgeschlagen welche andere Nutzer, die ebenfalls dieses Lied hörten, hören. Dieses Verfahren geht davon aus, dass durch die Verbindung der Lieder durch vorhergehende Aufrufe eine gute Aussage darüber getroffen werden kann wie gut diese Stücke zusammen passen. Werden Lieder häufig nach-

[4] Bewertungen eines Nutzers

[5] Beobachten des Konsumverhalten

einander gehört (todo), wird diese Verbindung höher bewertet und die Empfehlung häufiger ausgesprochen. Auch wird das Verhalten und der Musikgeschmack des Kunden selbst durch ein System analysiert, um so über Ähnlichkeiten der Kundenpräferenzen mit derer anderer, diesen wiederum bessere Empfehlungen aussprechen zu können. So werden Lieder einem Musikstil zugeordnet und so zielgerichtet dem Nutzer nahegelegt.
Verschiedene Studien ([7][8]) zeigen, dass KF alternative Methoden in der Genauigkeit übertrifft, weshalb es nicht nur im Bereich der Musikempfehlung als die erfolgreichste gilt.

2.5 Merkmalbasierter Filter

2.6 Kontextbasierter Filter

2.7 Hybride Methode

Bei hybriden Methoden werden kollaborative, merkmalsbasierte und kontextbasierter Filter miteinander verknüpft, wodurch ein besseres Empfehlungsergebnis mit weniger Nachteilen der einzelnen Methode zu erzielen. Meistens wird ein kollaborativer Filter mit einem der beiden anderem kombiniert.
Als *gewichtet* wird eine hybride Methode bezeichnet, bei der Empfehlungsrate der einzelnen Methoden durch eine Linearkombination zusammengerechnet wird. Das Ergebnis der Linearkombination stellt den Empfehlungswerte eines Liedes dar. Durch unterschiedliche Gewichtung der Methoden kann das Empfehlungsergebnis optimiert werden. Der *wechselnde* Ansatz benutzt ein bestimmtes Kriterium anhand dessen es die Methode zur Vorschlagbestimmung wechselt. Dies kann beispielsweise dann der Fall sein, wenn der erste Filter kein zuverlässiges Ergebnis[6] liefert. Dann wechselt das System den Filter und kann ein besseres Empfehlungsergebnis bekommen.
Bei *gemischten* hybriden Empfehlungen werden unterschiedliche Techniken[7] miteinander vermischt. Dadurch kann für ein System mit inhaltsbasierten Filter das Cold-Start Problem vermieden werden.
Hybride Methoden können einige Nachteile von kollaborativen Filtern entfernen. Allerdings stehen auch sie vor dem NUP. Dennoch sind hybride Methoden sehr beliebt, da Information über einen neuen Benutzer schnell herausgefunden[8] werden oder durch Profilangaben bereits nach der Registrierung vorhanden sind. [9]

3. Neuronale Netzen in der MIR

CNN sind durch das biologische Sehen inspiriert und konnten den ersten großen Erfolg im Bereich der Bildklassifizierung [10] verzeichnen. Trotzdem werden CNN auch in verschiedenen Audiobereich, wie der Spracherkennung [11] sowie in der MIR mehr genutzt und erforscht.
In der MIR nutzen die ersten Forschungen CNNs, um die Aufgabe der Musikgenre-Klassifizierung [12] zu untersuchen. Die Ergebnisse[9] zeigen, das eine automatisierte Klassifizierung die herkömmliche Methode MFCC deutlich übertrifft. Das erste CNN für inhaltsbasierte Musikempfehlung [2] benutzt zunächst eine Matrix-Faktorisierung um Merkmalsvektoren[10] für alle Lieder zu erhalten. Anschließend wird das Neuronale Netz für die Zuordnung der Audio-Inhalte an die Merkmalsvektoren genutzt. [6]

Im nachfolgenden Absatz werden die Schichten und das supervised[10] Training eines CNN beschrieben.

[6] semantische Unterschiede
[7] meist kollaborativ mit inhaltsbasiertem Filter
[8] Datamining
[9] richtigen Klassifizierung
[10] Eigenschaften eines Musters in Vektordarstellung
[10] Ausgabeergebnisse der Testdaten sind vorhanden

3.1 Convolutional Neuronalen Netze

In einer CNN Architektur werden drei Haupttypen von Schichten/Ebenen verwendet: Convolutional Layer (CL), Pooling Layer (PL) und Fully-Connected Layer (FCL). Jede Schicht besteht aus einer Anzahl von Knoten, die die Eingabedaten der Ebene wieder spiegeln. Knoten einer Schicht sind nur mit Knoten der nächsten Ebene verbunden. Diese Verbindung wird als Gewicht oder Parameter bezeichnet. Durch das Training von bekannten Daten und Ergebnissen werden diese Parameter automatisch angepasst. Anschließend ist das CNN fähig das Ergebnis unbekannter Daten zu errechnen.

3.1.1 Schichten eines Convolutional Neuronalen Netzwerks

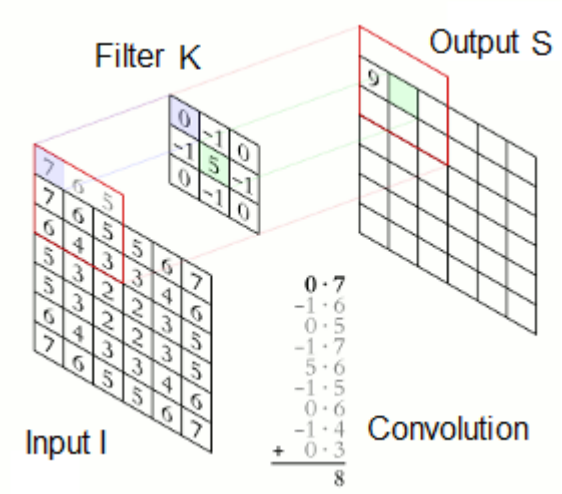

Abbildung 1: Faltung einer 6x6 Matrix mit einem 3x3 Filter [13]

Convolutional Layer

In einem CL findet eine Faltung der Eingangsdaten, in Form einer Matrix, und einem oder mehreren Filtern statt. Ein Filter dient beispielsweise zur Glättung oder zur Verkleinerung der Daten. Eine Verkleinerung der Eingangsmatrix findet statt, wenn ein Filter ohne Zeropadding[11] verwendet wird. Die Parameter eines Filters werden zufällig initialisiert, können aber mit Hilfe des Backpropagation Verfahren (3.1.2) angepasst werden. Werden mehrere Filter auf die Eingangsdaten angewendet, ändert sich die Tiefe der gesamten Ausgangsmatrix entsprechend der Anzahl der Filter. [14]
In Abbildung 1 ist die Eingabematrix I eine 6x6 Matrix und K ein 3x3 Filter. Die Ausgabematrix S wir an den Stellen (i,j) durch die nachfolgende Gleichung berechnet. Eine genauere Herleitung der Gleichung findet der Leser u. a. bei [15](328f).

$$S(i,j) = (I \star K)(i,j)$$

$$(I \star K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n)$$

Pooling Layer

Ein PL wird zwischen zwei CL eingefügt. Ihre Funktion besteht darin, die Größe der Daten zu reduzieren und damit die Anzahl der Parameter für das nächste CL. Durch die Reduzierung wird die Berechnung des gesamten Netzwerkes beschleunigt. [14]
Ein PL wandelt die Ausgabe eines CL, durch eine statistische Zusammenfassung von nebeneinander liegenden Ausgängen um. Verschiedene Methoden für ein Pl sind: Max Pooling [16], eine Übergabe der größten Zahl in einem rechteckigen Umfeld; die Durchschnittsberechnung des Umfeldes oder ein gewichteter Durchschnitt basierend auf der Entfernung eines zentralen Punktes [15](355).
Abbildung 2 zeigt einen 2x2 Max-Filter, der

[11] Eine Matrix wird am Rand um Nullen erweitert. Bsp. aus einer 7x7 Matrix wird eine 9x9 Matrix

auf eine 4x4 Datenmatrix angewandt wird. Die Verschiebung oder Stride des Filters ist 2 dh. der Filter wird zunächst auf der y-Achse verschoben. Erreicht er dort das Ende wird er um eine Stride auf der x-Achse verschoben und beginnt wieder mit der y-Verschiebung.

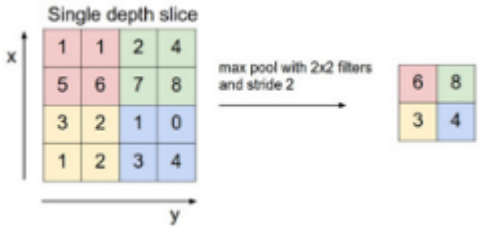

Abbildung 2: Maxpooling mit einem 2x2 Filter[14]

Fully-Connected Layer

Ein oder mehrere geschachtelte FCL dienen bei einem Klassifizierungsproblem aus Ausgabeschicht. Die Anzahl der Knoten in der letzten FCL entsprechen der Anzahl der Klassen, die das CNN unterscheiden soll. Jeder Knoten in einer FCL hat eine Verbindungen zu allen Knoten der vorherigen Schicht. Die Ausgabe der FCL ist ein Vektor, in der jeder Eintrag die Wahrscheinlichkeit der jeweiligen Klasse spiegelt. [14]

3.1.2 Training

CNNs werden durch die Backpropagation Methode trainiert. Backpropagation basiert auf dem Gradientenverfahren, welches versucht für die Fehlerfunktion E, durch sukzessive Iteration der Parameter ein globales Minimum zu finden, meistens aber nur ein lokales findet. Um das Minimum zu erreichen, werden die Werte der Gewichte w_{ij} durch Verwendung der Kettenregel, der partiellen Ableitung, berechnet:[12]

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}}$$

Der neue Wert eines Parameters w zum Zeitpunkt t lässt sich dann durch die folgende Formel berechnen:

$$w(t) = w(t-1) + \eta * \frac{\partial E}{\partial w}$$

Die Lernrate η ist eine Konstante die definiert werden muss.
Anhand der Kostenfunktion, kann die Parameteranpassung und damit der funktionale Wert eines CNN mit reellen Zahlen dargestellt und verglichen werden. Diese Funktion C lässt sich durch die Summe über Trainingsbeispiele m und einer Fehlerfunktion E, hier die negativ conditional Log-Likelihood[13] darstellen:

$$E(x,y) = -logp(y|x)$$

$$C = \frac{1}{m}\sum_{i=1}^{m} E(x^i, y^i)$$

Das Training eines Netzes ist abgeschlossen, wenn die Kostenfunktion minimal ist bzw. wenn in einem gegebenen Zeitraum keine bessere Kostenfunktion gefunden wird. [15](80ff,129ff)

3.2 Musikempfehlung mit Neuronalen Netzwerken

Anwendung eines CNN zur automatischen Musikempfehlung

Um das TODO Verzweigung TODO vorgestellt Problem der Semantischen Lücke zu lösen, kommt ein solches neuronales Netz zum Einsatz [2]. Dieser Ansatz wird wie folgt umgesetzt: das Netz wird darauf trainiert latente Faktoren aus einzelnen Musikstücken zu generieren, welche für eine Empfehlung verwendet

[12] Ausgabeparameter o, beliebige Anzahl an Knoten *net* zwischen Ausgabe und w

[13] Abgeleitet von Maximum Likelihood, Dichtefunktion für Maximafindung

werden. Dieses Verfahren wird anschließend im Vergleich mit einem konventionellen Ansatz, welcher dem Bag-of-Words Prinzip folgt. Daraus folgend wird beurteilt, inwiefern mittel CNN Eigenschaften zur spezifizierung des Nutzergeschmacks generiert und ausgewertet werden können. In den folgenden Kapiteln wird auf diesen Ansatz nun genauer eingegangen.

3.2.1 Datenbasis

Als Datenbasis zur Umsetzung dieses neuen Ansatzes wurden verschiedene Musik Datensätze in Betracht gezogen. Diese stehen im zusammenhang mit dem Million Song Dataset TODO ABkürzung TODO, dieser Datensatz verfügt über Metainformationen und bereits analysierter Audio Informationen von einer Millionen Liedern. Ebenso ist dieser Datensatz öffentlich zugänglich und kann kostenlos heruntergeladen werden, des weiteren stellt dieser derzeit die größte Forschungsdatenbasis im Gebiet der Musikanalyse dar. Zwei Datensätze im Umfeld des MSD sind von besonderem Interesse, der Echo Nest Taste Profile Subset Datensatz und der The Last.fm Datensatz. Die Problematiken mit diesen Datensätzen, waren zum einen einem die schlechte Dokumentation der Generierung der Informationen und der Daten selbst, sowie das keine unverarbeiteten Musikquellen mitgeliefert werden. Das Problem der nicht vorhanden Rohdaten konnte beseitigt werden, in dem für 99% des Datensatzes Musikschnipsel mit der Dauer von 29 Sekunden von 7digital.com bezogen wurden. Auf der Seite 7digital.com können einzelne Musikstücke 30 Sekunden lang Probegehört werden.TODO anders erläutern TODO TODO generell zu gleich mit dem Million Song Dataset Eintrag im Original Dokument, noch anpassen vll erweiter oder umändern TODO. Der ENTPS sticht durch seine Eigenschaft als größte, bereits mittels kollaborativen Filtern ausgwerteten, Informationsbasis hervor und bietet sich dadurch für ein Weiterverarbeitung mittels CNN an. Auch kann durch die größe der Daten eine realitätsnaher Versuch unternommen werden. Durch die Verwendung dieses Datensatzes, ist zu jedem Lied, pro Nutzer, die genaue Anzahl gespeichert, wie häufig das Stück vom Nutzer angehört wurde.

3.2.2 Weighted matrix facorization

Wie bereits im Kapitel TODO VERWEIS TODO erklärt wird beim Kooperativen Filter angenommen das ein Nutzer ein Lied nur dann oft hört, wenn es im auch gefällt. Um diese Daten für ein Training des neuronalen Netztes zu verwenden, muss ein spezieller Algortihmus angewandt werden. Die üblichen Algorithmen, welche für ein Training eines neuronalen Netzes verwendet werden, dessen ziel eine Errechnung von Bewertungen haben, wie TODO BEISPIEL nennen TODO können nicht auf Basis dieser Daten zum Einsatz kommen, denn sie können nicht verarbeiten wenn für ein Lied keine Wertung vorliegt. tODO Satz auftrennen TODO Der Umstand das ein Lied keine Wertung erhalten hat, kann mehrere Gründe haben, unter anderem das ein Nutzer dieses Lied schlicht und ergreifen nicht kennt. Der Nutzer könnte allerdings das Lied bereits kennen, aus anderen Quellen, es nicht mögen und deshalb diesen Titel nicht anhören. Dies führt zu zwei unterschiedlichen Szenarien auf Grundlage der gleichen Wertung. Um diesen Umstand richtig bewerten zu können, muss der Algorithmus flexibel sein. Deshalb kommt ein weighted matrix factorization Algorithmus zum Einsatz, um die Informationen aus dem Taste Profile Subset zu verarbeiten. Diese Idee wurde angelehnt an den Versuch bei der Bewertung von Fernsehshows und deren automatisierter Empfehlung dem Nutzer gegenüber möglichst gute Ergebnisse zu erzielen. Es wurde gespeichert wie oft ein Nutzer ein Fernsehformat angesehen hat. Auch hier tritt der Fall auf das für Formate von einzelnen Nutzern keine Bewertungen vorlagen. Dies konnte mehrere Gründe haben: der Nutzer kennt das Format nicht, eine

Sendung die er noch mehr schätzt kommt zur gleichen Sendezeit oder er mag die Sendung nicht. Folglich gibt es auch hier eine Ausgangswertung und drei Verschiedene Schlussfolgerungen sind möglich.

Dieser, für implizite Bewertungen optimierter, WMF Algorithmus ist wie folgend aufgebaut:

$$S(i,j) = (I \star K)(i,j) \quad (1)$$

$$(I \star K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n) \quad (2)$$

pui = I(Rui > 0), Cui = 1 + alphalog(1+Ehoch-1Rui)

Rui steht für die Anzahl an Wiedergaben je Nutzer pro Lied. Für jedes dieser Nutzer-Song Paare, wird eine Präferenz Variable Pui und eine Confidence Variable Cui definiert. Cui bildet hierbei die Indikatorfunktion, Alpha und Epsylon bilden Hyperparameter. U soll für Nutzer (User) und I für das Musikstück (Item) stehen. Die Präferenz Variable beschreibt wie häufig ein Nutzer U einen Track I angehört hat. Entspricht der Wert 1, gefällt ihm dieses Lied. Die Confidence Variable wird genutzt um abbilden zu können wie aussagekräftig die Bewertung eines Liedes ist. Ein hoher Wert ergibt sich, wenn das Lied häufig gespielt wurde, ein niedriger Wert dagegen wenn ein Lied selten gespielt wird.

Die Zielfunktion der WMF ist wie folgt gegeben:

TODO ZEICHEN TODO ist der regularization Parameter, Xu ist der latente Faktor. U steht auch hier wieder für den Nutzer. Yi stellt den Latenten Vector für Musikstück i dar.Es baut auf einen confidence-gewichted quadtratische Fehler Term, sowie auf L2 als regulization Term mean squared error TODO umschreiben klingt kacke TODO Term auf.

Beachten Sie, dass die erste Summe über alle Benutzer und alle Songs verteilt ist: im Gegensatz zur Matrix Faktorisierung für die Rating-Vorhersage, wobei die Begriffe den Benutzer-Element-Kombinationen entsprechen, für die Keine Bewertung verfügbar kann verworfen werden, wir müssen alle möglichen Kombinationen berücksichtigen. Wie Ein Ergebnis, bei dem ein stochastischer Gradientenabfall zur Optimierung verwendet wird, ist für einen Datensatz dieser Größe nicht praktikabel. Stattdessen wird eine effiziente Methode der alternierenden kleinsten Quadrate (ALS) verwendet, auch dies ist aus dem genannten Model angelehnt TODO CITE.

3.2.3 Extraktion und Analyse möglicher Faktoren aus dem Audiosignal

Um weitere Faktoren für eine Bessere Bewertung mit einfließen lassen zu können, müssen weitere Merkmale extrahiert und betrachtet werden. Die Vorhersage von Faktoren auf Grundlage von Audiosignalen eines Liedes lässt sich als Regressionsproblem definieren, die zu lernende Funktion muss eine Zeitreihe auf einen Vektor mit reellen Zahlen abbilden. Um dies zu lösen gibt es zwei Ansätze. Zum einen die Methode Bag-of-Words und das Training eines Convolutional Networks.

Um das bereits vorgestellte BoW Modell verwenden zu können, müssen die Audiofeatures aus den Lieder extrahiert, in eine BoW Darstellung überführt und aggregiert werden. Dadurch können, mittels klassischer Regressionstechnik, die Merkmale auf Faktoren abgebildet werden. Um einen Vergleich mit dem Neuronalen Netz ziehen zu können wurde dieses Modell wie folgt umgesetzt: 13 MFFCs wurden aus 1024 Audioschnipsel errechnet, des weiteren wurden unterschiede erster und zweiter Ordnung berechnet. Insgesamt wurden also 39 Koeffizienten errechnet. Mittels k-Means-Algorithmus zur Vektoreinquantisierung wurden 4000 Elemente gelernt und alle MFCC Vektoren dem nächsten Mittelwert zugeordnet. Anschließend wurde für jedes Lied gezählt, wie häufig ein Mittelwert ausgewählt wurde. Der daraus resultieren-

de Vektor, repräsentierte als BoW-Feature ein einzelnes Lied. Abschließend wurde mittels Hauptkomponentenanalyse diese Features verkleinert. Aus diesen Features entstanden durch Lineare Regression und einem mehrlagigen Perzeptron, einem vereinfachten Neuronalem Netz, die Faktoren zur automatischen Musikempfehlung.

Die zweite Möglichkeit ist es ein CNN zu trainieren. Faktoren welche durch den WMF, auf Basis der vorhanden Nutzerbewertungen, extrahiert werden, werden verwendet um die Vorhersagemodelle zu trainieren. Das WMF eignet sich hierfür besonders gut, da ein leistungsfähiges Optimierungsverfahren vorhanden ist. Um dieses Vorhaben umsetzen zu können, wurden 3 Erfolgsfaktoren definiert. Um das Netz gut trainieren zu können, ist es nötig auf eine große Menge an Trainingsdaten zurückgreifen zu können. Dies wird mittels der MSD, welche bereits vorgestellt wurde, abgesichert. Um das Problem das Vanishing Gradient Problem Todo erläutern und Verweisen TODO zu verrringern kommen rectified linear units zum Einsatz. Auch die Geschwindigkeit des Trainings an sich wird als kritisch betrachtet. Um ein schnelles Lernen zur ermöglichen, wurde der Lernvorgang parallelisiert und mittels GPU Unterstützung zur Berechnung optimiert. Eine GPU ist im Vergleich zu einer CPU in der Lage, in der gleichen Zeit, deutlich mehr Operationen durchzuführen, welche zum trainieren eines Neuronalen Netzes nötig sind. TODO GPU Unterstützung referenzieren TODO Das Trainining des CNN erfolgte in folgenden Schritten: Als erstes wurde wurden aus den Audiosignalen Zeit-Freuqzenz-Details extrahiert, um diese als Input für das Netz zu verwenden. Es handelten sich hierbei um Spektogramme, welche analog zum BoW-Modell aufgebaut waren. TODO satz überarbeiten TODO. Das Netz wurde mit 3 Sekundenlangen Audioschnipseln trainiert, welche zufällig aus den Songs entnommen wurden, dies sorgte für eine zusätzliche Beschleunigung der Trainingsgeschwindigkeit. Um die Features für den gesamten Song zu ermitteln wurden die Faktoren von aufeinanderfolgenden 3-Sekunden-Musikschnipseln gemittelt.

Da als Ergebniss der Zielfunktion die Faktoren als reelle Werte benötigt werden, muss der Quadratische Mittlere Fehler der vorhersagen reduziert werden.

Latent factor vectors are real-valued, so the most straightforward objective is to minimize the mean squared error (MSE) of the predictions. Alternatively, we can also continue to minimize the weighted prediction error (WPE) from the WMF objective function. Let yi be the latent factor vector for song i, obtained with WMF, and y0i the corresponding prediction by the model. The objective functions are then (represents the model parameters):

3.2.4 Versuchsaufbau und Durchführung vergleichender Tests

Nun werden die beiden Modelle, das BoW-Modell sowie das CNN-Modell, anhand deren Ergebnisse im Rahmen eines Experiments verglichen um die Leistungsfähigkeit des neuen Ansatzes bewerten zu können. Um die Qualität der Empfehlungen aber auch die extrahierten Faktoren an sich untersuchen zu können, wurden folgende drei Schritte unternommen.

Untersuchung der Vielfältigkeit der latenten Faktoren zu untersuchen werden diese mit Tags für Lieder aus einem Tag-Prediktion-Verfahren TODO Verweis oder Erklärung TODO verglichen. Tags können Songs beschreiben, unter anderem Genre, Instrumentierung, Tempo, Stimmung und Erscheinungsjahr. Dieser Vergleich wurde auf basis aller 9,330 Lieder des Datensatzes erstellt und die 50 Beliebtesten Tags aus der Last.fm Datenbank für jedes Stück extrahiert. Der vergleich ergab das die Vektoren einen erhöhte Empfehlungsgenauigkeit erbringen. tODO Verweis, stimmt das? TODo....

Mittels quantitative Evaluation wird untersucht wie gut aus den Audioquellen der Lieder Faktoren extrahiert werden können. Um

dies zu ermöglichen wurden die Faktoren verwendet um damit die ein Empfehlungssystem umzusetzen. Für jeden Nutzer u und jeden Song i der Datenbasis wurde eine Wertung xTuYi errechnet und das Lied mit dem höchten Wert als erstes vorgeschlagen. Zum Vergleich wurden ebenfalls für Nutzer und Songs mittels Bag-of-Words System Wertungen gefunden. In diesem Modell wurden alle Wertungen für einen gegebenen Nutzer in eine Durchnschnitts-Ähnlichkeits-Wertung umgerechnet basierend auf alle Lieder welche dieser Nutzer bereits hörte.

Um die Faktorvektoren vorhersagen zu können, wurde jeweils wie folgt verfahren: Eine Lineare Regressiont welche auf den Bag-Of-Words ansatz trainiert wurde. Ein MLP trainiert auf ebenfalls dem gleichen BoW-Modell. Ein CNN, welches auf log-skalierten Mel-Spektogrammen trainiert wurde, um den den Mittleren quadratischen Fehler der Vorhersagen zu reduzieren. Abschließend wurde das gleiche CNN trainiert um den WPE aus dem WMF zu verringern.

Für unsere ersten Experimente verwendeten wir eine Untergruppe der Datensatz, der nur die 9.330 beliebtesten Lieder enthält und Abhören von Daten für nur 20.000 Benutzer. Wir haben 1.881 benutzt Lieder zum Testen. Für die anderen Experimente haben wir verwendet alle verfügbaren Daten: Wir haben alle Lieder verwendet, die wir verwenden Daten für und dass wir einen Audioclip herunterladen konnten für (382,410 Lieder und 1 Million Benutzer insgesamt, 46.728 Lieder wurden zum Testen verwendet).

Wir geben die durchschnittliche Durchschnittsgenauigkeit an (mAP, abgeschnitten bei 500 Empfehlungen pro Benutzer) und der Bereich unter der ROC-Kurve (AUC) der Vorhersagen. Wir haben alles bewertet Modelle auf der Teilmenge, unter Verwendung von latenten Faktorvektoren mit 50 Dimensionen. Wir haben das Faltungsneuronal verglichen Netzwerk mit linearer Regression auf der Bag-of-Word-Darstellung auf dem gesamten Datensatz, unter Verwendung von Vektoren mit latenten Faktoren mit 400 Dimensionen. Ergebnisse sind in den Tabellen 2 bzw. 3 gezeigt.

Bei der Untergruppe scheint die Vorhersage der latenten Faktoren den metrischen Lernansatz zu übertreffen. Verwenden Ein MLP anstelle einer linearen Regression führt zu einer geringfügigen Verbesserung, aber die Einschränkung ist hier eindeutig die Textdarstellung der Wörter. Die Verwendung eines konvolutionellen neuronalen Netzwerks führt zu einem anderen große Leistungssteigerung. Wahrscheinlich liegt das daran, dass die Bag-of-Word-Darstellung nicht funktioniert reflektieren jede Art von temporaler Struktur.

Interessanterweise führt das WPE-Ziel nicht zu einer verbesserten Leistung. Vermutlich ist das so Die Gewichtung bewirkt, dass die Bedeutung der Lieder ihrer Popularität proportional ist. Im Mit anderen Worten, das Modell wird ermutigt, latente Faktorvektoren für populäre Lieder aus zu prognostizieren das Training lief sehr gut, auf Kosten aller anderen Songs.

Auf dem vollständigen Datensatz, der Unterschied zwischen den Beutelwörtern Ansatz und das konvolutionelle neuronale Netzwerk ist viel ausgeprägter. Beachten Sie, dass wir nicht trainiert haben MLP auf diesem Datensatz aufgrund der geringen Differenz in der Leistung mit linearer Regression auf der Teilmenge. Wir auch eingeschlossene Ergebnisse, wenn die latenten Faktorvektoren erhalten werden aus Nutzungsdaten. Dies ist eine obere Grenze für was ist erreichbar, wenn man sie aus dem Inhalt vorhersagt. Dort ist eine große Lücke zwischen unserem besten Ergebnis und diesem theoretischen Maximum, aber das ist zu erwarten: wie bereits erwähnt, viele Aspekte der Songs, die die Präferenz der Nutzer beeinflussen kann unmöglich nur aus Audiosignalen extrahiert werden. Insbesondere können wir die Popularität von die Lieder, die AUC und mAP erheblich beeinflussen Punkte.

Eine Bewertung der Qualität der gefundenen Faktoren, kann nicht nur auf der Betrach-

tung der Genauigkeit des Empfehlungssystems bezogen werden sondern sollten auch die vorgeschlagenen Lieder als Ergebniss betrachtet werden. Ein Vergleich der mittels CNN vorhergesagten Faktoren empfohlenen Liedern einerseits und die TODO welche Sinds Todo anderseits ergaben folgendes: Liedermengen sind sehr unterschiedlich, nur wenige Teilmengen vorhanden. Allerdings sind beide Ergebnissmengen ein gutes Ergebniss und die mittel CNN-Modell gefundenen Liedermenge ist etwas abwechslungsreicher was für ein Empfehlungssystem als Vorteil zu sehen ist.

3.3 Hybride Musikempfehlung mit einem Neuronalen Netzwerk

Im Unterschied zu der zuvor dargestellten Forschung (3.2) wird in der jetzigen ein Deep Belief Netzwerk(DBN) verwendet, um ein hybrides inhaltsbasiertes Musikempfehlungssystem zu entwickeln. Bisherige inhaltsbasiertes Systeme verfolgen typischerweise einem zweistufigen Ansatz: zunächst extrahieren sie aus Audioinhalte den MFCC Koeffizienten; anschließend prognostizieren sie Musikpräferenzen eines Nutzers. Das nachfolgende Modell führt dieses beiden Schritte simultan und automatisch aus. [6]
Das hybride Modell basiert auf einem hierarchischen linearen Modell mit einem Deep Belief Netzwerk(HLDBN), dass zunächst erläutert wird, um anschließend die Funktionsweise des hybriden Systems darzustellen.

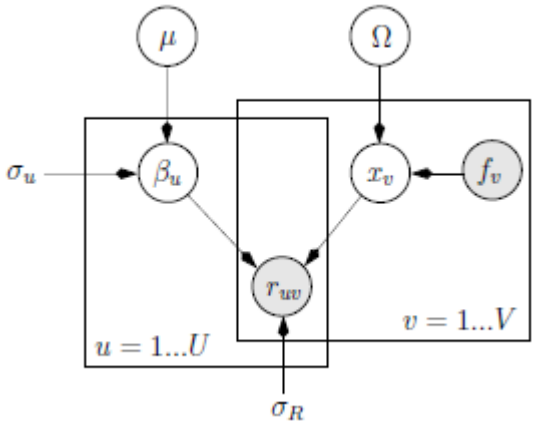

Abbildung 3: Hierarchisches lineares Modell eins Deep Belief Netzwerks [6]

3.3.1 Hierarchisches lineares Modell mit einem Deep Belief Netzwerk

Das in Abbildung 3 gezeigte Modell ist wie folgt definiert: f_v sind Musikmerkmale eines Liedes v, die durch den Merkmalsvektor x_v automatisch errechnet werden. Die bevorzugte Musik eines Benutzers u wird as Vektor β_u bezeichnet. Ω bezeichnet die Parameter, die das DBN lernt. Die Bewertung r_{xv}, die ein Nutzer einem Lied v gibt, ist ein Skalarprodukt von x_v und β_u. Durch σ_R wird die Varianz aller Bewertungen des Nutzers betrachtet. μ repräsentiert den allgemeinen Musikgeschmack aller Benutzer, wobei σ_u die Varianz des einzelnen Nutzers definiert. Alle Benutzer und Lieder Paare werden als I bezeichnet. Für eine Regularisierung der Werte wird die Gaußsche Normalverteilung $\mathcal{N}$ verwenden.[14] [6]
Das Modell ist wie folgt formuliert:

$$r_{xv} \sim \mathcal{N}(\beta' x_v, \sigma^2{}_R)$$

$$\beta \sim \mathcal{N}(\mu, \sigma^2{}_u I)$$

$$x_v = DBN(f_v; \Omega)$$

[14] $\mathcal{N}$(a,b) ist die Normalverteilung mit Mittelwert a und Varianz b. x $\sim$ p zeigt, dass x die Verteilung p erfüllt
[15] Ausgabeergebnisse der Testdaten sind nicht vorhanden
[16] nur ein kleiner Teil der vorhandenen Daten wird trainiert
[17] Spezialisierung

Für das Training des Systems wird im Unterschied zu einem CNN zunächst ein unsupervised[15] Training durchgeführt um die Knoten zu initialisieren. Anschließend findet ein supervised Training zur Optimierung der Parameter statt. Als Optimierungsmethode wird das stochastische Mini-Batch[16] Verfahren mit Backpropagation genutzt, um ein Overfitting[17] des Modells zu vermeiden. Nach der Lernphase kann die Bewertung r_{xv} eines Benutzers u über ein Lied v geschätzt werden, wodurch diesem neue Lieder empfohlen werden können. [6]

3.3.2 Hybrides Modell mit einem Deep Belief Netzwerk

Basiert auf dem HLDBM wird das in Abbildung 4 gezeigte Modell um einen kollaborative Filter erweitert, um eine noch bessere Empfehlungsrate zu erhalten.

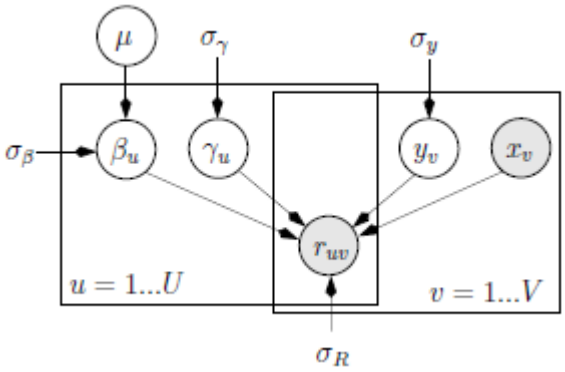

Abbildung 4: Hybrides Empfehlungs Modell [6]

Die Musikmerkmale eines Liedes x_v und die bevorzugte Musik eines Nutzers β_u werden wie in (3.4.1) berechnet. γ_u stellt einen Merkmalsvektor eines Benutzers u; y_v den Merkmalsvektor eines Liedes v dar. β_u, γ_u und y_v werden durch das Neuronale Netzwerk anhand der Trainingsdaten gelernt. Die Empfehlungsrate r_{uv} ergibt sich aus der Summe der Skalarmultiplikationen ${\gamma'}_u y_v$ und ${\beta'}_u x_v$. Die A-priori-Wahrscheinlichkeit[18] werden durch die folgenden Formeln definiert:

$$r_{xv}|\beta_u, x_v, \gamma_u, y_v, \sigma_R \sim \mathcal{N}({\beta'}_u x_v + {\gamma'}_u y_v, {\sigma_R}^2)$$
$$\beta_u|\sigma_\beta \sim \mathcal{N}(\mu, |{\sigma_\beta}^2 I)$$
$$\gamma_u|\sigma_u \sim \mathcal{N}(0, |{\sigma_\gamma}^2)$$
$$y_v|\sigma_v \sim \mathcal{N}(0, |{\sigma_y}^2)$$

Die Fehler Funktion des Modells wird auch mit Hilfe der Backpropagation minimiert. Alternative können die Werte für γ_u und y_v durch eine PMF berechnet und als Initialwerte verwendet werden. [6]

4. Vergleich der vorgestellten Modelle

Abschließend wird nun ein Vergleich zwischen dem vorgestellten CNN Ansatz einerseits und dem danach folgendem DBN gezogen. In Versuchen [2] wurde festgestellt das dass CNN Model einem BOW System überlegen ist und eine bessere Empfehlungsrate erreicht. Das anschließend erläuterte Versuchsmodel, welches mittels DBN einen hybriden Methode verfolgt, konnte im direkten Vergleich zu CNN eine nochmals verbesserte Empfehlungsgenauigkeit erreichen. Vergleichende Versuchsreihen [6] haben folgendes festgestellt: ein nicht hybrider Ansatz welcher allein auf Training der beiden Netze ergab, dass das vorgestellte HLDBN Modell genauere Ergebnisse lieferte als das vorgestellte CNN. Die integration der beiden Modelle in einen hybriden Aufbau ergab wiederum ebenso dass der Einsatz des HLDBN Netzes zu genaueren Empfehlungsraten führte als die Verknüpfung von CNN und CF. Die Empfehlungsraten für bereits bekannte Lieder, konnten somit im Vergleich zu klassischen Ansätzen

[18] Anfangswahrscheinlichkeit

verbessert werden. Auch das bereits eingeführte Problem des NSP konnte durch den Einsatz Neuronaler Netze gelöst werden. Sowohl der Einsatz des CNN als auch der des HLDBN Netzes führen hierbei zum Erfolg. Sowohl der im Punkt 3.2 vorgestellte ansatz mittels eines CNNs sowie der in Punkt 3.3 erläuterte Ansatz mit Einsatz eines hybriden dbn Netzwerkes eine gute Möglichkeit zur automatisierten Musikempfehlung. Beide Verfahren haben in Rahmen von Versuchen bewiesen, dass sie sowohl zuverlässig sind, aber auch das sie einen klassisches Verfahren wie ein BoW-System

Literatur

[1] Joshua P. Friedlander. News and notes on 2017 mid-year riaa revenue statistics. *RIAA*, 2017.

[2] Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. *Advances in Neural Information Processing Systems 26*, 2013.

[3] Òscar Celma. *Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer, 2010.

[4] Markus Schedl, Arthur Flexer, and Julián Urbano. *The neglected user in music information retrieval research*, volume 36. Springer, 2013.

[5] Peter Knees and Markus Schedl. *Music Similarity and Retrieval*, volume 41. Springer, 2016.

[6] Xinixi Wang and Ye Wang. Improving content-based and hybrid music recommendation using deep learning. *Proceedings of the ACM International Conference on Multimedia*, pages 627–636, 2014.

[7] B. McFee, T. Bertin-Mahieux, D. P. W. Ellis, and G. R. G. Lanckriet. The million song dataset challenge. *21st International Conference Companion on World Wide Web*, pages 909–916, 2012.

[8] Luke Barrington, Reid Oda, and Gert Lanckriet. Smarter than genius? human evaluation of music recommender systems. *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pages 357–362, 2009.

[9] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact*, pages 331–370, 2002.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] Alex Graves, Abdel-Tahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *Acoustics, Speech and Signal Processing, IEEE International Conference on*, pages 6645 – 6649, 2013.

[12] Honglak Lee, Yan Largman, Peter Pham, and Andrew Y. Ng. Unsupervised feature learning for audio classi

cation using convolutional deep belief networks. *Advances in Neural Information Processing Systems*, 2009.

[13] wikipedia, 2017. `https://de.wikipedia.org/wiki/Convolutional_Neural_Network#/media/File:3D_Convolution_Animation.gif`.

[14] Andrej Karpathy. *Convolutional Neural Networks for Visual Recognition*. Stanford University, 2017. `https://github.com/cs231n/cs231n.github.io/blob/master/convolutional-networks.md#conv`.

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[16] Zhou Y. and Chellappa R. Computation of optical flow using a neural network. *IEEE International Conference*, 71–78, 1988.