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

Ein wachsender Streamingkonsum und die immer größer werdenden Musikauswahl, der Streamingplattformen, erfordern für ein angenehmes Nutzungserlebniss, bessere automatisierte Musikempfehlungsysteme. In dieser Arbeit werden Grundlagen der automatisierten Musikempfehlung und deren klassische Ansätze erläutert. Aufbauend auf diese wird, das Maschinellen Lernen, mit neuronalen Netzen, als nächster Schritt zu einem verbessertem Empfehlungsergebnis betrachtet und ein Vergleich zweier verschiedener Methoden gezogen.

1. Einleitung

Im ersten Halbjahr des Jahres 2017 wurden 62% der Einnahmen der amerikanischen Musikindustrie durch Streaming Plattformen[1] erzielt. Im Vergleich zum Vorjahr erhöhten sich dadurch die Einnahmen um 48% auf 2.5$ Milliarden [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 Stilen 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. Das selbe Problem gibt es bei einem neuen Benutzer: Diesem kann kein adäquater Vorschlag gemacht werden, da es an Informationen mangelt, welche Art von Musik ihm gefällt. [3]
Um diese Probleme zu lösen und die allgemeine Empfehlungsqualität zu steigern werden auch Neuronale Netze, wie das Convolutional Neuronalen Netzwerk (CNN), als Grundlage für solch Empfehlungssysteme erforscht und erprobt.

Der weitere Verlauf dieser wissenschaftlichen Arbeit ist wie im folgt organisiert: Im 2. Abschnitt werden die Grundlagen der Musikempfehlung vorgestellt. Im 3. Kapitel wir der Aufbau von Neuronale Netze, sowie zwei Neuronale Netze für inhaltsbasierte Musikempfehlung beschrieben. Diese Arbeit, schließt

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

im letzten Abschnitt, mit einem Vergleich, der Ergebnisse der beiden Modelle.

2. Grundlagen der Musikempfehlung

In dem Gebiet der 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, die Melodie, die Harmonie oder die Stimmung eines Stückes.
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 sich 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 der Nutzer 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, um sie für Empfehlungen an den Nutzer zu verarbeiten. [6]
Im Folgenden Abschnitt werden zunächst die Begriffe, Mel Frequency Cepstral Coefficients (MFCC) und Bag of Words (BOW) erklärt. Anschließend werden die verschiedenen Filtermethoden erläutert.

2.1 Mel Frequency Cepstral Coefficients

Die Mel Frequency Cepstral Coefficients werden zur Analyse, von Musikstücken, verwendet, um Metadaten zuordnen zu können. Mel steht für die wahrgenommenen Tonhöhen, eines Liedes. Durch dieses Verfahren können die Tonhöhen, getrennt von der Sprache, betrachtet werden. Für die Analyse eines Liedes sind besonders die Tonhöhen von Bedeutung. Diese Verfahren wird auch für Spracherkennung genutzt, hier ist allerdings die gesprochene Sprache wichtiger als die Tonhöhe. [7]

2.2 Bag of Words

Bag of Words 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. [8]
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. Für eine Klasse wird definiert welche Features für diese von spezifischer Natur sind, danach können die Lieder zugeordnet werden. [9]

2.3 Kollaborativer Filter

Kollaboratives Filtern prognostiziert die Vorlieben eines Hörers, indem es aus unterschiedlichen Benutzer-Lied-Verhältnissen lernt. Es

[4] Bewertungen eines Nutzers

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 [10].
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 nacheinander gehört, 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 die 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.[10]

2.4 Merkmalbasierter Filter

Mittels des merkmalsbasiertem Filters, werden Nutzern Musikstücke aufgrund aus Lieder gewonnener Informationen vorgeschlagen. Dies bedeutet im Detail, dass aus den Musikstücken, mittels verschiedenster Metriken, die Audio Signale eines Liedes analysiert werden um Erkenntnisse über die Stimmung eines Musikstücks, die Frequenz oder Rhythmus zu erhalten. Auf Grund dieser Informationen können Stücke, dem Konsumenten, vorgeschlagen werden, die einen gleichen oder sehr ähnlichen Inhalt bieten. [12]

2.5 Kontextbasierter Filter

Kontextbasiertes Filtern verwendet Informationen, um Lieder zu beschrieben und zu charakterisieren. Diese Informationen können Metainformationen, wie etwa Genre, Emotionen, Tags und Labels sein. Auch Erscheinungsjahr, Künstler und Album werden hierfür zur Bewertung zu Rande gezogen. Es werden nach Ähnlichkeiten der gehörten Lieder des Nutzer mit noch ungehörten verglichen. Auf Basis der dadurch entstandenen Bewertungen werden dem Nutzer Lieder empfohlen. [12]

2.6 Hybride Methoden

Bei hybriden Methoden werden kollaborative, merkmalsbasierte und kontextbasierter Filter miteinander verknüpft. Dadurch kann ein besseres Empfehlungsergebnis mit weniger Nachteilen der einzelnen Methode erzielt werden. Meistens wird ein kollaborativer Filter mit einem der beiden anderen 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 Empfehlungswert 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

[5] Beobachten des Konsumverhalten

[6] semantische Unterschiede

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. [13]

3. Neuronale Netze in der MIR

CNN sind durch das biologische Sehen inspiriert und konnten den ersten großen Erfolg, im Bereich der Bildklassifizierung [14], verzeichnen. Auch deshalb werden CNN auch in verschiedenen Audiobereich, wie der Spracherkennung [15], sowie in der MIR mehr genutzt und erforscht.
Erste Forschungen im Bereich der MIR nutzen CNNs, um die Aufgabe der Musikgenre-Klassifizierung [16] zu untersuchen. Die Ergebnisse[8] zeigen, dass eine automatisierte Klassifizierung die herkömmliche Methode, MFCC, deutlich übertrifft. Das erste CNN für inhaltsbasierte Musikempfehlung [2], benutzt zunächst eine Matrix-Faktorisierung, um Merkmalsvektoren[9], 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.

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 Neuronale Netzwerks

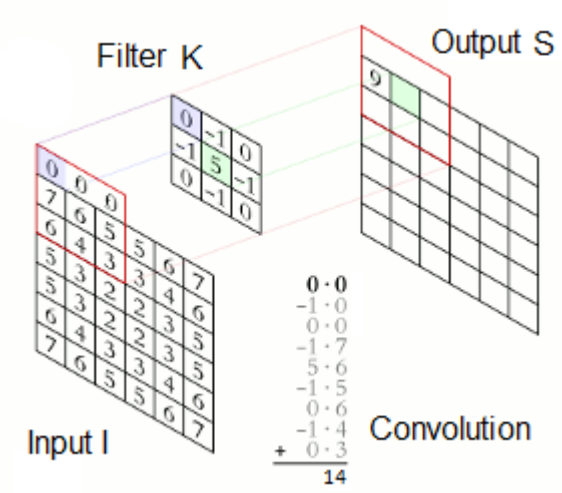

Abbildung 1: Faltung einer 6x6 Matrix (mit Zero-padding) und einem 3x3 Filter [17]

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 Zero-padding[11] verwendet wird. Die Parameter eines Filters werden zufällig initialisiert, können aber, mit Hilfe des Backpropagati-

[7] meist kollaborativ mit inhaltsbasiertem Filter
[8] richtige Klassifizierung
[9] Eigenschaften eines Musters in Vektordarstellung
[10] Ausgabeergebnisse der Testdaten sind vorhanden
[11] Eine Matrix wird, bei einer Faltung, am Rand, um Nullen erweitert. Bsp. aus einer 7x7 Matrix wird eine 9x9 Matrix

on Verfahrens, vgl. Kapitel 3.1.2, angepasst werden. Werden mehrere Filter auf die Eingangsdaten angewendet, ändert sich die Tiefe der gesamten Ausgangsmatrix entsprechend der Anzahl der Filter. [18]
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 [19](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. [18]
Ein PL wandelt die Ausgabe eines CL, durch eine statistische Zusammenfassung von nebeneinander liegenden Ausgängen, um. Verschiedene Methoden für ein Pl sind: Max Pooling [20], eine Übergabe der größten Zahl in einem rechteckigen Umfeld; die Durchschnittsberechnung des Umfeldes oder ein gewichteter Durchschnitt, basierend auf der Entfernung eines zentralen Punktes [19](355).
Abbildung 2 zeigt einen 2x2 Max-Filter, der 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.

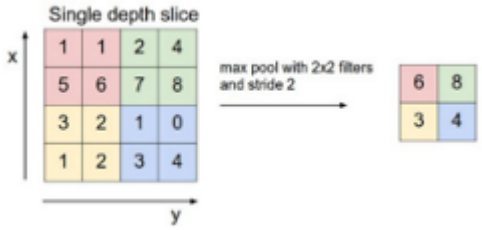

Abbildung 2: Maxpooling mit einem 2x2 Filter[18]

Fully-Connected Layer

Ein oder mehrere geschachtelte FCL dienen als Ausgabeschicht. Jeder Knoten in einer FCL, hat eine Verbindung, zu allen Knoten, der vorherigen Schicht. Abhängig von der Problemstellung, Klassifizierung oder Regression, ist die Anzahl der Ausgabeknoten unterschiedlich. Bei einer Klassifizierung entspricht die Anzahl, in der letzten FCL, der der Klassen, die das CNN unterscheiden soll. Die Ausgabe der FCL ist ein Vektor in der jeder Eintrag, die Wahrscheinlichkeit, der jeweiligen Klasse, spiegelt. Bei einer Regression gibt das FCL einen oder mehrere Realwerte aus. [18]

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]:

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

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

Der neue Wert eines Parameters w, zum Zeitpunkt t, lässt sich durch 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:

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

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

Das Training eines Netzes ist abgeschlossen, wenn die Kostenfunktion minimal ist bzw. in einem gegebenen Zeitraum, keine bessere gefunden wird. [19](80ff,129ff)

3.2 Anwendung eines CNN zur automatischen Musikempfehlung

Dieser Ansatz wird wie folgt umgesetzt: das Netz wird darauf trainiert latente Faktoren, also Eigenschaften, aus einzelnen Musikstücken zu generieren, welche für eine Empfehlung verwendet werden. Dieses Verfahren wird anschließend im Vergleich mit einem konventionellen Ansatz, welcher dem Bag-of-Words Prinzip folgt, betrachtet. Daraus folgend wird beurteilt, inwiefern mittel CNN Eigenschaften zur Spezifizierung des Nutzergeschmacks generiert und ausgewertet werden können.

3.2.1 Datenbasis

Als Datenbasis zur Umsetzung dieses neuen Ansatzes wurden Datensätze des Million Song Dataset (MSD) verwendet, der Echo Nest Taste Profile Subset (ENTPS), sowie der The Last.fm Datensatz. Das MSD verfügt über Metainformationen und bereits analysierter Audio Informationen von Liedern. Ebenso ist dieses öffentlich zugänglich und kann kostenlos heruntergeladen werden. Des weiteren stellt dieses derzeit die größte Forschungsdatenbasis im Gebiet der Musikanalyse dar. Da bei diesen Datensätzen keine Roh-Musikdaten mitgeliefert werden, wurden für die einzelnen Lieder jeweils 29 Sekunden Ausschnitte von der Seite 7digital.com genutzt.[21] [2]

3.2.2 Weighted matrix factorization

Der ENTPS Datensatz enthält, die Abspielhäufigkeit eines Songs für jeden Nutzer, also eine implizite Bewertung des Nutzer für diese Lieder. Allerdings kann für ein Musikstück, für einen oder mehrere Nutzer, auch keine Bewertung vorliegen. Dieser Umstand, dass 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 verwendete Algorithmus flexibel sein. Deshalb kommt ein weighted matrix factorization (WMF) Algorithmus zum Einsatz, um die Informationen aus dem ENTPS trainieren zu können. [2] Diese Idee ist angelehnt an den Versuch bei der Bewertung von Fernsehshows und deren automatisierter Empfehlung, dem Nutzer gegenüber, möglichst gute Ergebnisse zu er-

[13] Abgeleitet von Maximum Likelihood, Dichtefunktion für Maximafindung

zielen. 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 kann 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. [22]
Dieser, für implizite Bewertungen optimierte, Algorithmus ist wie folgt aufgebaut:

$$p_{ui} = I(r_{ui} > 0),$$
$$c_{ui} = 1 + \alpha \log(1 + \epsilon^{-1} r_{ui})$$

r_{ui} ist die Anzahl wie häufig ein Lied $_{\text{i}}$ durch den Nutzer $_{\text{u}}$ gehört wurde. Für jedes dieser Datenpaare, wird eine Präferenz Variable p_{ui} definiert. Sie indiziert ob der Nutzer das Lied jemals angehört hat, entspricht der Wert, der Variable, 1 dann wird ausgegangen der Nutzer mag das Lied. Die zweite Variable c_{ui}, zeigt an wie stark die Bewertung des Liedes in das Empfehlungssystem einfließen soll. Lieder mit höheren Spielraten, werden stärker berücksichtigt, der Wert von c_{ui} ist hoch, da man sich hier sicherer ist dass, der Nutzer, das Lied mag. Lieder mit geringen Spielraten werden mit einem niedrigen Wert für c_{ui} bedacht. α und ϵ sind Hyperparameter.

Die Zielfunktion der WMF ist wie folgt gegeben:

$$\min_{x_\star, y_\star} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u ||x_u||^2 + \sum_i ||y_i||^2 \right),$$

λ ist der Regularations-Parameter. X_u stellt den Eigenschafts Vektor des Nutzers $_{\text{u}}$ dar. y_i den Vektor für das Musikstück $_{\text{i}}$ dar. Die Zielfunktion besteht aus einem L2 Regularisationsterm und einem Term über die Mittlere Quadratische Abweichung für die bereits erläuterten Variablen. Die Summe im ersten Teil der Funktion beinhaltet jede mögliche Kombination von Nutzern und Liedern, deshalb kommt eine Methode zur Näherung der alternierenden kleinsten Quadrate zum Einsatz. Dieser Ansatz ist ebenfalls an das bereits genannte Verfahren, zur Empfehlung von Fernsehsendungen.[2]

3.2.3 Extraktion und Analyse möglicher Faktoren aus dem Audiosignal

Der Akt der Extraktion von Eigenschaften auf Basis von Audiosignalen eines gegeben Liedes, ist ein Regressionsproblem. Dies benötigt das Lernen einer Funktion welche Zeitintervalle auf einen Vektor reeller Zahlen abbilden kann. Um dies zu Lösen, wird ein neuer Ansatz verfolgt: die Nutzung eines CNNs. Die mittels des, berteis erläuterten, WMF Algorithmus gefunden Vektoren auf Basis der vorhanden Daten, werden eingesetzt um das Empfehlungssystem zu trainieren. 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. 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.

Das Trainining des CNN erfolgte in folgenden Schritten: Als erstes wurden aus den Audiosignalen Zeit-Freuqzenz-Details extrahiert, um diese als Input für das Netz zu verwenden Dies erfolgte durch logarithmisch komprimierten Mel-Spektogramme welche aus 128 Komponenten bestehen. 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 Ergebnis der Zielfunktion die Faktoren als reelle Werte benötigt werden, muss der Quadratische Mittlere Fehler der vorhersagen reduziert werden. Alternativ wurde auch eine Verbesserung der Genauigkeit der WMF Zielfunktion betrachtet. [2]

3.2.4 Versuch und Durchführung vergleichender Tests

Um die Leistungsfähigkeit des neuen Ansatzes auf Basis eines neuronalen Netzes darlegen zu können, wurde ein Experiment mit diesem durchgeführt. Auch wurde ein klassisches Modell, basierend auf dem bereits erläuterten BoW Ansatz umgesetzt, um die Ergebnisse vergleichen zu können. Um die Qualität der Empfehlungen aber auch die extrahierten Faktoren an sich untersuchen zu können, wurden folgende Schritte unternommen.

Um die Vielseitigkeit der extrahierten Audio-Features zu untersuchen, wurden diese, mit Tags für Lieder aus einem Tag-Prediktion-Verfahren verglichen. Tags können Songs beschreiben, unter anderem Genre, Instrumentierung, Tempo, Stimmung und Erscheinungsjahr. Dieser Vergleich wurde auf Basis aller 9,330 Lieder des Last.fm Datensatzes erstellt und die 50 Beliebtesten Tags für jedes Lied extrahiert. Der Vergleich ergab das die Vektoren einen erhöhte Empfehlungsgenauigkeit erbringen.
Um quantitativ zu beurteilen, wie gut die aus den Audioquellen der Lieder den Liedern extrahiert Faktoren werden können wurden die Vorhersagen verwendet um damit diesem für ein Empfehlungssystem umzusetzen. Für jeden Nutzer $_u$ und jeden Song textsubscripti der Datenbasis wurde eine Wertung $x^T{}_u y_i$ errechnet und das Lied mit dem höchten Wert als erstes vorgeschlagen. Zum Vergleich wurde ebenfalls eine Metrik auf Basis von Liederähnlichkeiten mittels Bag-of-Words System gelernt um Vorschlagsbewertunge zu erhalten. 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 Vektoren zur Empfehlung generieren zu können, wurde jeweils Folgende Ansätze verfolgt: Eine Lineare Regression welche auf den Bag-Of-Words Ansatz trainiert wurde. Ein mehrlagigen Perzeptron, ein vereinfachtes künstliches Netz, trainiert auf ebenfalls das gleiche BoW-Modell. [23] Ein CNN, welches auf log-skalierten Mel-Spektogrammen trainiert wurde, um den den Mittleren quadratischen Fehler der Vorhersagen zu reduzieren. Sowie das gleiche CNN trainiert um den Empfehlungen aus dem WMF zu verbessern. [2]

In ersten Experimenten wurden nur Teilmengen der vorhanden Datensätze genutzt. Betrachtet man die Durchschnittsgenauigkeit der Empfehlungssystem ergibt sich folgendes Bild: Die lernenden Ansätze sind Leistungsfähiger als die Lineare Regression. Selbst der Ansatz basieren auf dem Perzeptron erweist sich als besser, als das Regressionsmodell. Die beiden auf CNN basierenden Modelle sind deutlich besser im Vergleich zu den beiden erstgenannten Modellen. Die vergrößerung der Datenbasis für weitere Experimente lässt die Leistungsfähigkeit und den Leistungsvorsprung der beiden auf CNN basierenden Ansätze noch größer werden. [2]

Eine Bewertung der Qualität der gefundenen Vektoren, kann nicht nur durch Metriken bestimmt werden. Ein Vergleich der mittels CNN vorhergesagten Vektoren empfohlenen Liedern einerseits und empfehlungen eines 50-dimesionalen WMFs anderseits ergaben folgendes: die empfohlenen Lieder sind zumeist unterschiedlich, nur wenige Teilmengen vorhanden. Allerdings erbringen beide Modelle ein gutes Ergebnis und die mittel CNN-Modell vorgeschlagenen Auswahl an Liedern ist etwas

abwechslungsreicher was für ein Empfehlungssystem als Vorteil zu sehen ist. [2]

3.3 Hybride Musikempfehlung mit einem Neuronalen Netzwerk

Im Unterschied, zu der davor dargestellten Forschung in Kapitel 3.2, wird nun ein Deep Belief Netzwerk (DBN) verwendet, um ein hybrides inhaltsbasiertes Musikempfehlungssystem zu entwickeln. Im Gegensatz zu einem CNN, nutzt ein DBN ein so genanntes Pretraining[14] als Parameterinitialisierung. 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 diese beiden Schritte simultan und automatisch aus. [6]
Das hybride Modell basiert auf einem hierarchisch linearen Modell mit einem Deep Belief Netzwerk (HLDBN), das zunächst erläutert wird, um anschließend die Funktionsweise, des hybriden Systems, darzustellen.

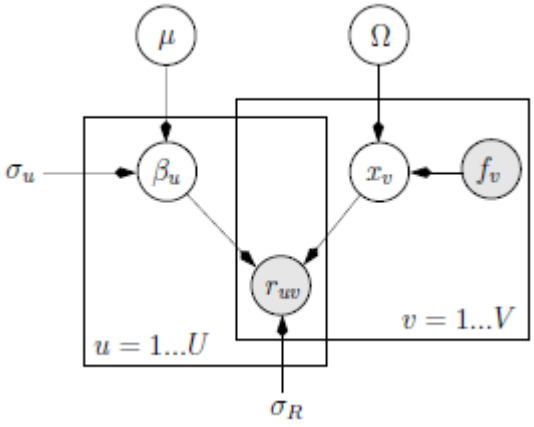

Abbildung 3: Hierarchisches lineares Modell eins Deep Belief Netzwerks [6]

3.3.1 Hierarchisch 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 als 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.[15]
Das Modell ist wie folgt formuliert:

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

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

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

Für das Training des Systems wird im Unterschied zu einem CNN zunächst ein unsupervised[16] Training durchgeführt um die Knoten zu initialisieren. Anschließend findet ein supervised Training zur Optimierung der Parameter statt. Als Optimierungsmethode wird das stochastische Mini-Batch[17] Verfahren mit Backpropagation genutzt, um ein Overfitting[18] 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]

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

3.3.2 Hybrides Modell mit einem Deep Belief Netzwerk

Basierend auf dem HLDBM, wird das in Abbildung 4 gezeigte Modell, um einen kollaborative Filter erweitert, wodurch eine noch bessere Empfehlungsrate erreicht wird.

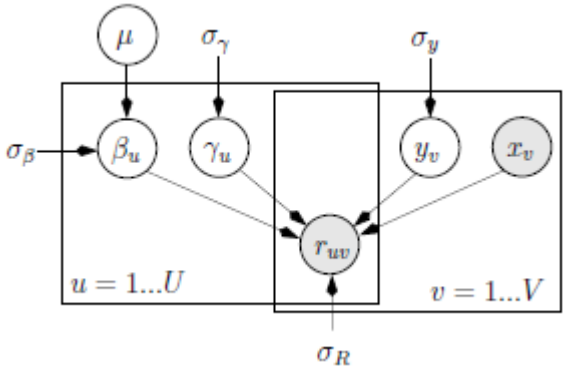

Abbildung 4: Hybrides Empfehlungs Modell [6]

Die Musikmerkmale eines Liedes x_v wird wie in Kapitel 3.4.1 berechnet. Die bevorzugte Musik eines Nutzers, wird als β_u bezeichnet. Der kollaborative Teil: γ_u stellt einen Vektor über alle Benutzer u dar; y_v den Vektor über alle Lieder v. β_u, γ_u und y_v werden gemeinsam, 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-Wahrscheinlichkeiten[19] 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. Um Zeit zu sparen, können die Werte für γ_u und y_v durch eine PMF berechnet und als Initialwerte verwendet werden. [6]
Um Informationen aus einem Lied, als Eingabeparameter für das DBN, nutzen zu können, müssen diese in einen Vektor umgewandelt werden. Dies geschieht in verschiedenen Schritten. In [6] werden alle Lieder in ein .wav Dateiformat, mit einer 8 kHz Abtastrate und 16 Bit Tiefe, umgewandelt. Aus den einzelnen Liedern wird ein 5 Sekunden Abschnitt genutzt, der in ein 166x120 Spektrogramm umgewandelt wird. Anschließend wird eine Hauptkomponentenanalyse[20] durchgeführt, wodurch sich ein Vektor für den Liedabschnitt ergibt.

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 basiert, 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 führt, als die Verknüpfung von CNN und CF. Die Empfehlungsraten für bereits bekannte Lieder, konnten somit im Vergleich zu klassischen Ansätzen verbessert werden. Auch das bereits eingeführte Problem des NSP konnte

[19] Anfangswahrscheinlichkeiten

[20] Zusammenhang der Darstellung einer Menge an Variablen durch wenige Faktoren

durch den Einsatz Neuronaler Netze gelöst werden. Sowohl der Einsatz des CNN als auch der des HLDBN Netzes führen hierbei zum Erfolg. Das ebenfalls bereits vorgestellte NUP konnte dagegen nicht gelöst werden, da in desem Modellen derzeit keine Möglichkeit besteht aus einem, dem System unbekannten, Nutzer Merkmale für eine Vektordarstellung zu generieren. Während aus einem Lied, durch die vorgestellten Modelle Audiofeatures extrahiert und dem System zugeführt werden können. Sowohl der in Kapitel 3.2 vorgestellte Ansatz mittels eines CNNs, sowie der in Kapitel 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 und durch bessere Empfehlungsraten einem klassischen Ansatz überlegen sind.

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] Beth Logan. Mel frequency cepstral coefficients for music modeling. *Cambridge Research Laboratory*, 2000.

[8] Deepu S., Pethuru Raj, and S. Rajaraajeswari. A framework for text analytics using the bag of words (bow) model for prediction. *International Journal of Advanced Networking and Applications (IJANA)*, pages 320–323, 2013.

[9] Chang-Hsing Lee, Hwai-San Lin, and Ling-Hwei Chen. Music classification using the bag of wors model of modulation spectral features. *15th International Symposium on Communications and Information Technologies (ISCIT)*, 2015.

[10] 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.

[11] 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.

[12] Juuso Kaitila. A content-based music recommender system. Master thesis, University of Tmpere, 2017.

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

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural net-

works. *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] 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.

[16] 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.

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

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

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

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

[21] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011.

[22] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, 2008.

[23] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review Vol. 65, No. 6*, 1958.