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 große Anzahl an verschiedenen Stile und Genres, beeinflussen sowohl soziales- und geographisches Umfeld, sowie der aktuelle Gemütszustandes 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. [7]
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, sowei 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 [3] 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 Stücks.
Als *Musikkontext* versteht man alle Aspekte, die nicht aus dem Audiosignal abgeleitet werden,

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

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. [4]

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 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 [7]. 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 [8].
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 (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 ([8][9]) zeigen, dass KF alternative Methoden in der Genauigkeit übertrifft, weshalb es nicht nur im Bereich der Musikempfehlung als die erfolgreichste gilt.

2.2 Merkmalsbasierter Filter

2.3 Kontextbasierter Filter

2.4 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 Empfehlungswertes 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 Vorschlagsbestimmung wechselt. Dies kann bei-

[4] Bewertungen eines Nutzers
[5] Beobachten des Konsumverhalten
[6] semantische Unterschiede

spielsweise 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. [10]

3. Neuronale Netzen in der MIR

CNN sind durch das biologische Sehen inspiriert und konnten den ersten großen Erfolg im Bereich der Bildklassifizierung [11] verzeichnen. Trotzdem werden CNN auch in verschiedenen Audiobereich, wie der Spracherkennung [12] sowie in der MIR mehr genutzt und erforscht.
In der MIR nutzen die ersten Forschungen CNNs, um die Aufgabe der Musikgenre-Klassifizierung [13] zu untersuchen. Die Ergebnisse[9] zeigen, das eine automatisierte Klassifizierung die herkömmliche Methode MFCC deutlich übertriff. Das erste CNN für inhaltsbasierte Musikempfehlung [2] benutzt zunächst eine Matrix-Faktorisierung um Eigen Vektoren für alle Lieder zu erhalten. Anschließend wird das Neuronale Netz für die Zuordnung der Audio-Inhalte zu den Eigen Vektoren genutzt. [6]

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

[7] meist kollaborativ mit inhaltsbasiertem Filter
[8] Datamining
[9] richtigen Klassifizierung

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

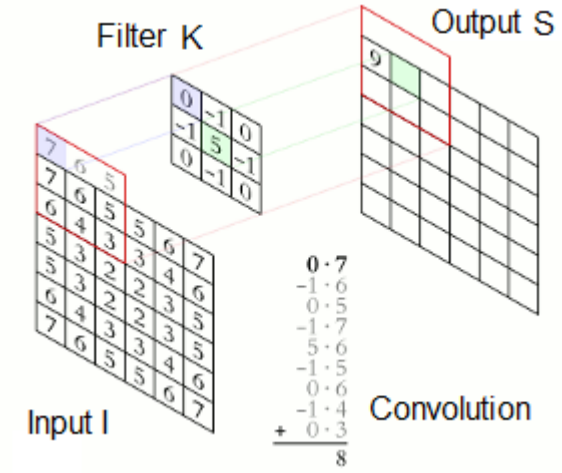

Abbildung 1: Faltung einer 6x6 Matrix mit einem 3x3 Filter [16]

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 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 [17], 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 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.

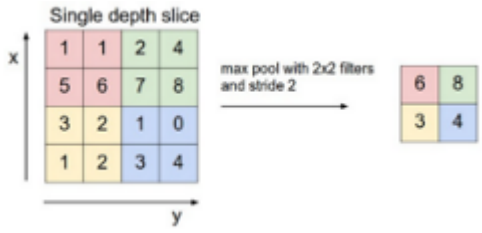

Abbildung 2: Maxpooling mit einem 2x2 Filter[14]

Fully-Connected Layer

toDo
Neuronen in einer FCL haben Verbindungen zu allen Knoten der vorherigen Schicht. Ihre Aktivierung wird durch eine Matrixmultiplikation und einem Bias-Offset berechnet [14]. Die FCL wird als Ausgabeschicht verwendet um aus der Eingangsmatrix einen Vektor zu erzeugen.

3.1.2 Supervised 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 Parameter w_{ij} durch Verwendung der Kettenregel, der partiellen Ableitung, berechnet:[12]

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

[10] Ausgabeergebnisse der Testdaten sind vorhanden
[11] Eine Matrix wird am Rand um Nullen erweitert. Bsp. aus einer 7x7 Matrix wird eine 9x9 Matrix

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

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.

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

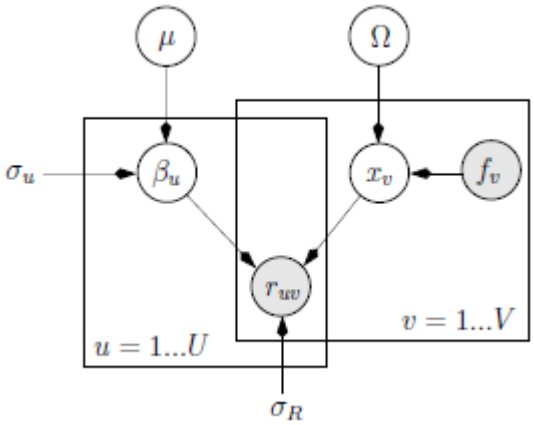

Abbildung 3: Hierarchisches lineares Modell eins Deep Belief Netzwerks [6]

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 Eigenvektor x_v automatisch errechnet werden. Die bevorzugte Musik eines Benutzer u wird as Vektor β_u bezeichnet. Ω bezeichnet die Parameter, die das DBNs lernt. Die Bewertung, die u einem Lied v gibt, ist eine Skalarprodukt von r_{xv} 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:

$$r_{xv} \sim \mathcal{N}(\beta' x_v, \sigma^2{}_R)$$
$$\beta \sim \mathcal{N}(\mu, \sigma^2{}_u I)$$
$$x_v = DBN(f_v; \Omega)$$

[13] Abgeleitet von Maximum Likelihood, Dichtefunktion für Maximafindung

[14] $\mathcal{N}$(a,b) ist die Normalverteilung mit Mittelwert a und Varianz b. x $\sim$ p zeigt, dass x die Verteilung p erfüllt

Für das Training des Systems wird im Unterschied zu einem CNN zunächst ein unsupervised[15] Training durchgeführt um die Knoten zu initialisieren. Als Optimierungsmethode wird das stochastische Mini-Batch[16] Verfahren mit Backpropagation genutzt, um ein Overfitting[17] des Modells zu vermeiden. Nach der Lernphase kann r_{xv} geschätzt werden, wodurch auch neue Lieder empfohlen werden können. [6]

3.3.2 Hybrides Modell mit einem Deep Belief Netzwerk

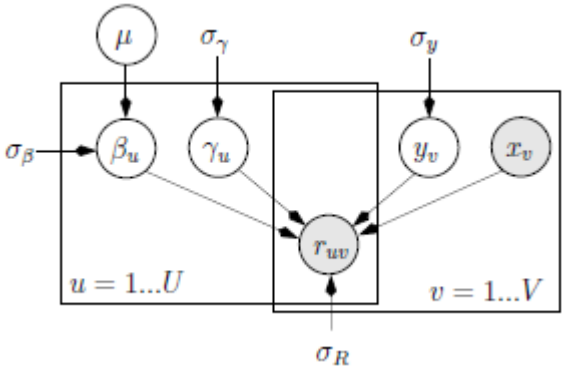

Abbildung 4: Hybrides Empfehlungs Modell [6]

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 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

[15] Ausgabeergebnisse der Testdaten sind nicht vorhanden

[16] nur ein kleiner Teil der Daten wird trainiert

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

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

[5] Gabriel Vigliensoni and Ichiro Fujinaga. Automatic music recommendation systems: Do demographic, pro

ling, and contextual features improve their performance? *Proceedings of the 17th International Society for Music Information Retrieval Conference*, pages 94–100, 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] Òscar Celma. *Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer, 2010.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

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

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

[17] Spezialisierung