

Neuronale Netze in der Videoproduktion

Laura Anger

Technische Hochschule Köln
Institut für Medien- und Phototechnik
laura.anger@th-koeln.de

Vera Brockmeyer

Technische Hochschule Köln
Institut für Medien- und Phototechnik
vera.brockmeyer@smail.th-koeln.de

Zusammenfassung

Vera

Schlüsselwörter

Faltungsnetze, Videoproduktion

Vera

1. Einleitung

Vera

2. Grundlagen

Laura

In Kapitel 2.1 wird zunächst ein allgemeiner Überblick über die verschiedenen Arbeitsschritte einer Videoproduktion. Im drauf folgenden Kapitel werden die Grundlagen von NNs zusammengefasst. Um dann in Kapitel 2.3 vertiefend auf Faltungsnetze einzugehen, deren Training in Kapitel 2.4 zusammengefasst wird.

2.1. Videoproduktion

Laura

Mit der Videoproduktion oder auch Filmproduktion wird die Herstellung sowohl von Kino- als auch von Werbe- und Fernsehfilmen zusammengefasst. In Abbildung 1 ist ein Ablaufplan einer typischen Videoproduktion zu sehen. Da es alleine in Deutschland über 850 Produktionsformen gibt [11] (Stand 2014),

kann der Ablaufplan nur einen sehr allgemeinen Überblick über die notwendigen Arbeitsschritte bieten.

Der erste Schritt, die Konzeption soll sowohl die Projektentwicklung, als auch die Vorproduktion zusammenfassen. Die sich anschließende Produktionsphase kann grob, wie im Schaubild zu sehen in Bild und Ton unterteilt werden, wobei diese beiden Bereiche nicht immer getrennt betrachtet werden sollten. Die Postproduktion besteht aus vielen verschiedenen Arbeitsschritten, deren Schwerpunkt auf dem Schnitt und der digitalen Bildnachbearbeitung liegt. Der Schritt der Distribution ist hier der Vollständigkeit halber erwähnt und beschreibt die Filmverwertung.

Der Aufbau der folgenden Ausführungen orientiert sich an Abbildung 1 (vgl. Kapitel 3, 4 und 5).

2.2. Neuronale Netze

Laura

NNs finden unter anderem Anwendung bei der Steuerung von Robotern, Börsenkursanalysen, Medizin oder Fahrzeugsteuerung. In der Bildverarbeitung werden NNs vor allem zur Klassifizierung genutzt.

Sie sind vom menschlichen Gehirn inspiriert, welches laut [4] ein nicht-lineares, komplexes und hochparalleles System zur Verarbeitung von Informationen darstellt. Ähnlich wie dieses bestehen künstliche NNs aus einer Menge an simulierten Neuronen, die untereinander verbunden sind und in Schichten organisiert sind. Es gibt verschiedene Arten der Vernetzung, die,

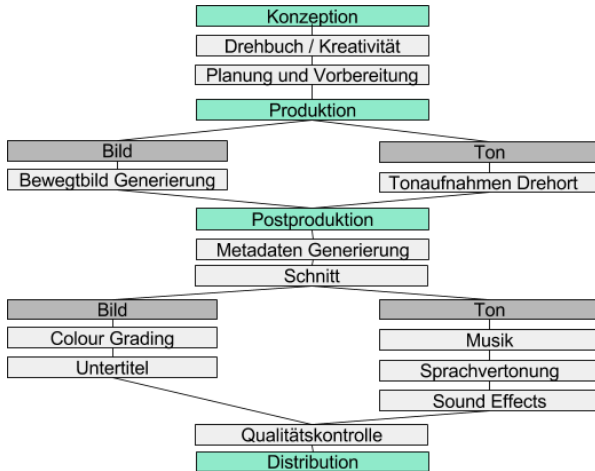


Abbildung 1. Grundlegende Arbeitsschritte einer Videoproduktion.

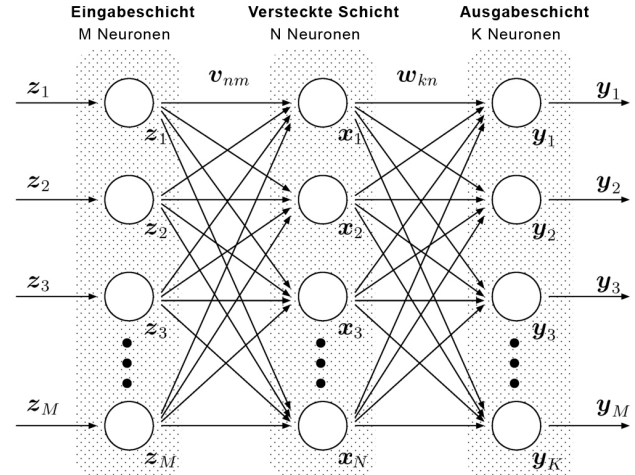


Abbildung 2. Prinzipieller Aufbau MLP nach [5].

wie in [4] und [10] beschrieben, in rück- und vorwärts gekoppelte Modelle unterteilt werden können.

Am häufigsten kommen sogenannte *Multilayer Perceptrons* (MLP) [1][8][10] zum Einsatz. Wie der Name vermuten lässt, werden hierbei die Neuronen in Schichten angeordnet. Ein solcher Aufbau ist beispielhaft in Abbildung 2 zu sehen. Dieses MLP besteht aus einer Eingabe- und Ausgabeschicht mit M bzw. K Neuronen und einer versteckten Schicht mit N Neuronen. Es handelt sich um ein vorwärtsgekoppeltes Modell, bei welchem jedes Neuron einer Schicht mit jedem Neuron der darauffolgenden Schicht verbunden ist. Dies nennt man volle Verbindung. Die Eingangsschicht dient zum Verteilen der Eingangswerte z_m mit $m = 1, \dots, M$. Die Ausgabe eines jeden Neurons in der versteckten Schicht, dargestellt durch x_n , lässt sich durch Formel 1 berechnen. Hierbei steht v_{nm} für die jeweilige Gewichtung der Verbindungen zwischen den Neuronen der Eingabe- und der versteckten Schicht und f für die Aktivierungsfunktion [10][4] des jeweiligen Neurons.

$$x_n = f \left(\sum_{m=1}^M v_{nm} z_m \right) \quad (1)$$

Die Ausgangswerte y_k , mit $k = 1, \dots, K$, lassen sich äquivalent unter Hereinnahme der Werte x_n und der Gewichte w_{kn} , sowie einer Aktivierungsfunktion g

berechnen, und gelten als Vertrauenswerte. Sie müssen gemäß der Aufgabenstellung interpretiert werden.

2.3. Faltungsnetze

Laura

Im Folgenden wird genauer auf Faltungsnetze eingegangen, da diese die Grundlage, für die meisten der in den folgenden Kapiteln vorgestellten Ansätze, bilden. Vereinfacht ausgedrückt besteht ein Faltungsnetz aus einer Vernetzung von Faltungsoperationen mit unterschiedliche Filtermasken. Faltungsnetze kommen, bedingt durch ihre Architektur, oft zum Einsatz, wenn große Datenmengen von einem NN verarbeitet werden sollen. Ein schematischer Aufbau ist in Abbildung 3 zu sehen.

Jedes Pixel eines Feldes, das auf der Abbildung zu sehen ist, wird durch ein Neuron repräsentiert. Die Felder sind in Schichten organisiert. Die Eingangsschicht fungiert, vergleichbar wie bei den MLPs aus Kapitel 2.2, als Verteiler der Information an die Neuronen der nächsten Schicht S_1 . Die Besonderheit eines Faltungsnetzes sind die sich abwechselnd durchgeführte Faltung und anschließende Unterabtastung. Zwischen den Schichten S_4 und S_6 ähnelt das Modell einem MLP, da die Neuronen schichtweise voll verbunden sind.

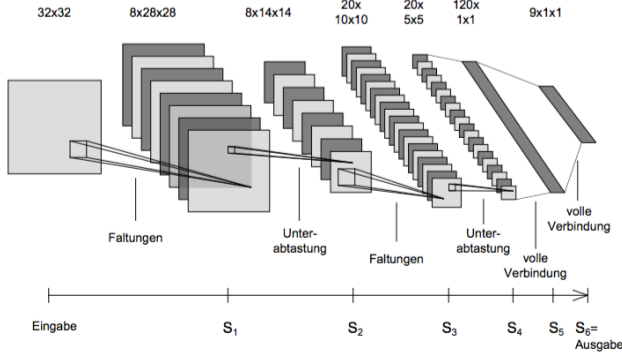


Abbildung 3. Prinzipieller Aufbau Faltungsnetz nach [9].

Im Allgemeinen wird für eine Faltung eine Filtermaske h , also eine endlicher zweidimensionaler Koeffizientensatz, wie in Formel 2 zu sehen, verwendet. Hierbei stehen x und y jeweils für die horizontale bzw. die senkrechte Bildkoordinate. Die Anzahl der Koeffizienten a_{xy} , wird in der Horizontalen mit N_{hx} und im Vertikalen mit N_{hy} bezeichnet.

$$h(x, y) = \begin{cases} 0 & \text{für } x < -\lfloor \frac{N_{hx}-1}{2} \rfloor \vee y < -\lfloor \frac{N_{hy}-1}{2} \rfloor \\ 0 & \text{für } x > \lfloor \frac{N_{hx}-1}{2} \rfloor \vee y > \lfloor \frac{N_{hy}-1}{2} \rfloor \\ a_{xy} & \text{sonst} \end{cases} \quad (2)$$

Formel 3 beschreibt die Faltung eines Eingangssignals s mit einer Filtermaske h , wobei I das Ausgangssignal in Abhängigkeit von x und y beschreibt.

$$I(x, y) = (s * h)(x, y) = \sum_{m_x=-\infty}^{\infty} \sum_{m_y=-\infty}^{\infty} s(m_x, m_y) \cdot h(x - m_x, y - m_y) \quad (3)$$

Im Fall eines Faltungsnetzes wird die Faltung, die wie in Abbildung 3 zu sehen, beispielsweise zwischen der Eingangsschicht und S_1 vollzogen wird, durch die Verbindung zwischen den Neuronen zweier Felder modelliert. Dabei entsprechen die Gewichte der Neuronen genau den Filterkoeffizienten a_{xy} . Für ein jedes Feld sind diese Koeffizienten konstant, was bedeutet, dass alle Neuronen eines Feldes mit nur einem Gewicht auskommen. Dieses Prinzip nennt man geteilte Gewichte.

Im Faltungsnetz wird nach jeder Faltung eine Unterabtastung durchgeführt um zu gewährleisten, dass die Dimension der Eingangsdaten schrittweise an die Dimension des Ausgangsvektors angepasst wird. Hierzu wird meist eine bilineare Unterabtastung um den Faktor 2 vorgenommen. Allgemeiner betrachtet werden $n \times n$ Werte zu einem Wert zusammengefasst.

Wie zu Beginn des Kapitels erwähnt, haben Faltungsnetze gegenüber den MLPs den Vorteil, dass sie nahezu beliebig hochskaliert werden können und somit gut geeignet für große Datenmengen sind. Dies liegt vor allem daran, dass die Neuronen nur lokal verbunden sind und sich somit das Prinzip der geteilten Gewichte zu Nutze gemacht werden kann. Ein weiterer Vorteil von Faltungsnetzen, der vor allem in der Bildverarbeitung genutzt wird, ist das sie translationsinvariant sind.

2.4. Training Faltungsnetze

Laura

Meistens werden Faltungsnetze mittels der *back-propagation* Methode trainiert. Bei dieser überwachten Lernmethode bedarf es einer großen Menge an vorher klassifizierten Eingabematerialien [7].

In den Faltungsschichten kann der Fehler der vorangegangenen Schicht nach Formel 4 berechnet werden. Dabei steht E für den Fehler in der jeweiligen Schicht l . Während x^l für die Eingabe in die Schicht steht, bezeichnet y^l die Ausgabe der entsprechenden Schicht. Die Größe der Eingabe wird der Einfachheit halber als quadratisch, also $m \times m$ -groß angenommen. Eine Gewichtung wird mit w bezeichnet. für Um die Formel in der Realität anzuwenden, muss die linke und obere Grenze des Eingabeinhaltes, z.B. eines Bildes, mit Nullen ergänzt werden. Ansonsten wäre es nicht möglich den Fehler für Pixel zu berechnen, welche näher als m an den entsprechenden Rändern liegen.

$$\begin{aligned} \frac{\delta E}{\delta y_{ij}^{l-1}} &= \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \frac{\delta E}{\delta x_{(i-a)(j-b)}^l} \frac{\delta x_{(i-a)(j-b)}^l}{\delta y_{ij}^{l-1}} \\ &= \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \frac{\delta E}{\delta x_{(i-a)(j-b)}^l} w_{ab} \end{aligned} \quad (4)$$

Weil alle Gewichtungen mittels des *back-propagation* Algorithmus während des Trainings angepasst werden, können Faltungsnetze laut LeCun als Erzeuger ihrer eigenen Merkmalsextraktion gesehen werden [6].

Vera

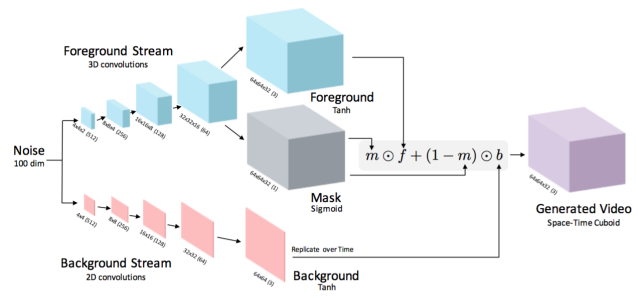
Vera

Laura

In diesem Kapitel werden zunächst Ansätze vorgestellt, die auf Grundlage von NNs Arbeitsschritte bei der Produktion von Videos übernehmen bzw. vereinfachen könnten (vgl. Kapitel 4.1). Dazu ist an zu merken, dass diese Ansätze meist in einem anderen Kontext entwickelt wurden und ggf. eine Anpassung an die Standards einer Produktion stattfinden müsste.

Laura

An der *University of Washington* ist im Rahmen einer Projektarbeit ein Musik Generator namens *Algo*



Rythm entstanden [12]. Für die Umsetzung habend die Studierenden um Timmerman RNNs geeignet trainiert. Der Quellcode kann auf Github [13] eingesehen werden.

Der letzte Ansatz benutzt ebenfalls RNNs um auf Grundlage einer Notensequenz ein Musikstück zu komponieren [2]. Dabei wird die Eingabesequenz zunächst interpretiert. Anschließend sorgen zwei RNN basierte Algorithmen für die Produktion von sowohl Rhythmus, als auch die Vorhersage der nächsten Note.

Alle drei Ansätze weisen hohes Potential auf, wenn es darum geht Musikstücke automatisiert zu generieren. Über das Genre des zugeführten Musikmaterials lässt sich die gewünschte Ausgabe begrenzt steuern. Es wäre durchaus vorstellbar, so Musik für eine Filmproduktion zu generieren.

Laura

Laura

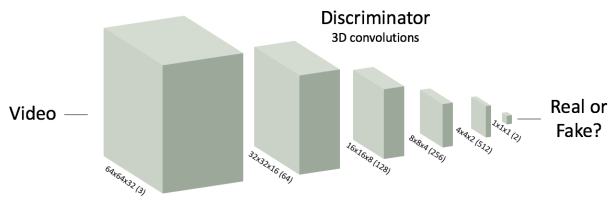


Abbildung 5. Diskriminator nach [14].

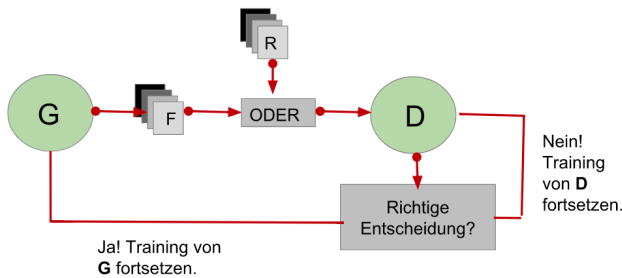


Abbildung 6. Diskriminator und Generator.

5. Postproduktion

Vera

5.1. Aktueller Stand

Vera

6. Zusammenfassung

Vera

Literatur

- [1] H. Braun. *Neuronale Netze Optimierung durch Lernen und Evolution*. Springer, 1997.
- [2] C. Browne. System and method for automatic music generation using a neural network architecture, Oct. 2 2001. US Patent 6,297,439.
- [3] D. Eck and J. Lapalme. Learning musical structure directly from sequences of music. (1300), 2008.
- [4] S. Haykin. *Neural Networks A Comprehensive Foundation*. Macmillan College Publishing Company, 1994.
- [5] T. Isokawa, H. Nishimura, and N. Matsui. Quaternionic multilayer perceptron with local analyticity. *Information*, 3(4):756, 2012.
- [6] Y. LeCun and Y. Bengio. The handbook of brain theory and neural networks. chapter Convolutional Networks for Images, Speech, and Time Series, pages 255–258. MIT Press, Cambridge, MA, USA, 1998.
- [7] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 253–256, May 2010.
- [8] D. Nauck, F. Klawonn, and R. Kruse. *Neuronale Netze und Fuzzy Systeme*. Vieweg, 1994.
- [9] M. Osadchy, Y. LeCun, and M. Miller. Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research*, 8:1197–1215, 2007.
- [10] J. Stanley and E. Bate. *Neuronale Netze Computersimulation biologischer Intelligenz*. Systhema Verlag GmbH, 1991.
- [11] statista. Anzahl der aktiven film- und fernsehproduktionsfirmen in deutschland in den jahren 1998 bis 2014. <https://de.statista.com/statistik/daten/studie/243238/umfrage/anzahl-der-film-und-fernsehproduktionsfirmen-in-deutschland/>. Abgerufen: 10.08.2016.
- [12] G. Timmermann. Algo rhythm: Music composition using neural networks. <https://medium.com/@granttimmerman/algo-rhythm-music-composition-using-neural-networks-f89897ff2df7>. Abgerufen: 18.08.2016.
- [13] G. Timmermann. Algorithmic music composition using artificial neural nets, howpublished = <https://github.com/grant/algo-rhythm>, note = Abgerufen: 18.08.2016,.
- [14] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. *CoRR*, abs/1609.02612, 2016.