# 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 [12] (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 [5] ein nicht-lineares, komplexes und hoch paralleles 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,

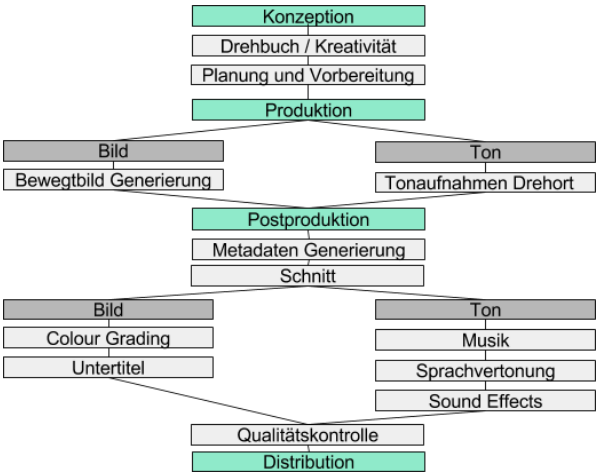


**Abbildung 1:** Grundlegende Arbeitsschritte einer Videoproduktion.

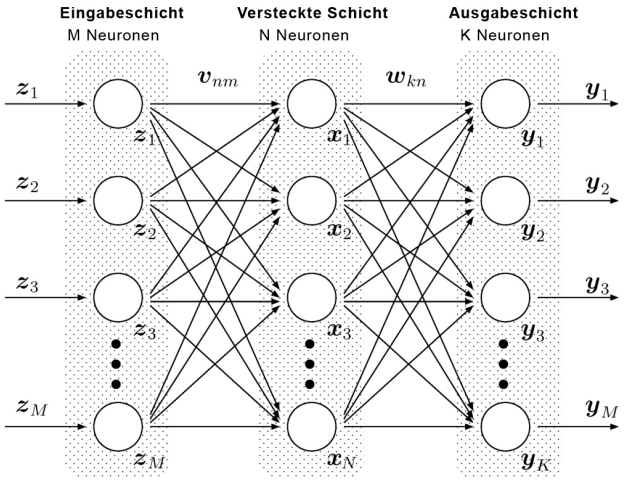


**Abbildung 2:** Prinzipieller Aufbau MLP nach [6].

wie in [5] und [11] beschrieben, in rück- und vorwärts gekoppelte Modelle unterteilt werden können.

Am häufigsten kommen sogenannte *Multilayer Perceptrons* (MLP) [1][9][11] 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 eine 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 Schichte verbunden

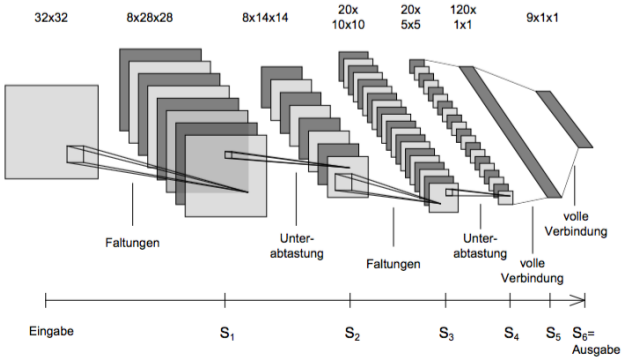


**Abbildung 3:** Prinzipieller Aufbau Faltungsnetz nach [10].

ist. Dies nennt man volle Verbindung. Die Eingangsschicht dient zum Verteilen der Eingangswerte $z_m$ mit $m = 1, ..., M$. 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 Aktivierungfunktion [11][5] des jeweiligen Neurons.

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

Die Ausgangswerte $y_k$, mit $k = 1, ..., K$, 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.

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.

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

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

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

Im Fall eines Faltungsnetzes wird die Faltung, die wie in Abbildung 3 zu sehen, beispielsweise zwischen der Eingangsschicht und $S_1$ vollzogen wird, 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 $nxn$ 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 [8].
In den Faltungsschichten kann der Fehler der vorangegangen 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 $mxm$-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.

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

Die Schichten, in denen die Unterabtastung stattfindet, leisten kaum Beitrag zum eigentlichen Lernprozess des Faltungsnetzes. Hier wird das Problem allerdings reduziert, da $nxn$ Werte in einem einzigen resultieren.

Weil alle Gewichtungen mittels des *back-propagation* Algorithmus während des Trainings angepasst werden, können Faltungsnetze laut LeCun als Erzeuger ihrer eigenen Merkmalextraktion gesehen werden [7].

# 3. Vorverarbeitung

Vera

## 3.1. Aktueller Stand

Vera

# 4. Produktion

Laura

muss angepasst werden

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.

In Kapitel 4.2 wird ein Ansatz zur automatisierten Generierung von Szenendynamiken in Hinsicht auf Funktionsweise und Arbeitserleichterung für die Videoproduktion analysiert.

## 4.1. Aktueller Stand Musikgenerierung

Laura

Es gibt verschiedene Ansätze NNs zu nutzen, um Musik automatisiert generieren zu lassen. Im Folgenden werden drei Ansätze kurz vorgestellt, welche alle rückgekoppelte Neuronale Netze (RNN) benutzen.
An der *University of Washington* ist im Rahmen einer Projektarbeit ein Musik Generator namens *Algo Rythm* entstanden [13]. Für die Umsetzung habend die Studierenden um Timmerman RNNs geeignet trainiert. Der Quellcode kann auf Github [14] eingesehen werden.
Ein weiterer Ansatz, der in [3] beschrieben wird, arbeitet ebenfalls mit einem RNN, welches mit einem Autokorrelation-basierte Prädiktor kombiniert wird. Dabei soll die Struktur von Musikstücken erlernt werden, indem zunächst die folgende Note einer Tonreihenfolge vorausgesagt werden soll.
Der letzte Ansatz benutzt ebenfalls RNNs um auf Grundlage einer Notensequenz eine 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.

## 4.2. Szenendynamik nach Vondrick et al.

Laura

Mit dem Ansatz von Vondrick et al. [15] können aus Einzelbildern ganze Szenen Dynamiken erstellt werden, welche zum einen für die Klassifizierung und zum anderen für die Vorhersage von Bewegung genutzt werden kann. Das Resultat sind kleine Videos mit einer Auflösung von 65x64 Pixeln und 32 Einzelbildern.

Die Grundlage für dieses Verfahren bilden zwei Faltungsnetze, welche als Gegenspieler trainiert werden. *Generative Adversarial Networks* (GAN) [4] Dieses Konzept basiert auf einem Generator und Diskriminator und w gegeneinander Trainieren siehe Formel (1)
unsupervised learning. Trainiert nach Kategorien

## 4.3. Bewertung des Ansatzes von Vondrick

Laura

Vorteile Verfahren: - automatisierte Vorverarbeitung des Lernmaterials möglich - Vorverarbeitung muss nur einmal gemacht werden - Trainingsmaterial für G kann auch für D genutzt werden - Bewegungen werden nach Kategorien gelernt
Vorteile in Bezug auf Videoproduktion: - Verständnis der Szenen Dynamik wird wichtiger

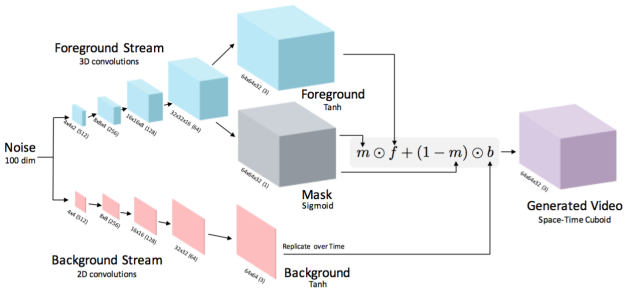


**Abbildung 4:** Aufbau Generator nach [15].

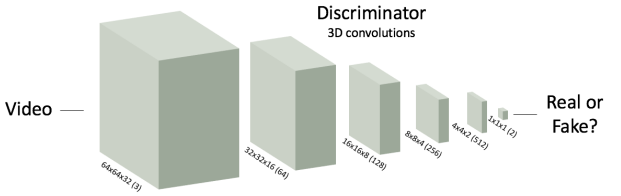


**Abbildung 5:** Aufbau Diskriminator nach [15].

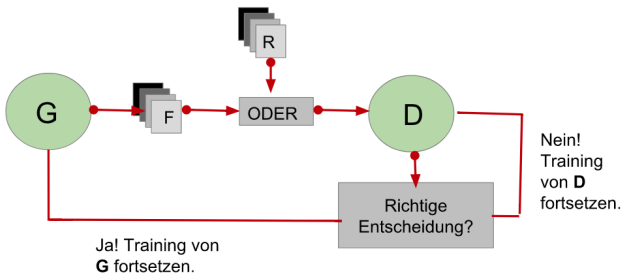


**Abbildung 6:** Zusammenspiel Diskriminator und Generator.

- planen höhere Auflösung - Denkbar für Realisierung von bspw. GIFs - Simulationen oder Vorhersagen

Nachteile: - bedarf viel Trainingsmaterial - Kategorien ohne Vernunft ausgewählt - Realismusgrad nicht ausreichend - Es gibt noch keine höhere Auflösung

# 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] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[5] S. Haykin. *Neural Networks A Comprehensive Foundation*. Macmillan College Publishing Company, 1994.

[6] T. Isokawa, H. Nishimura, and N. Matsui. Quaternionic multilayer perceptron with local analyticity. *Information*, 3(4):756, 2012.

[7] 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.

[8] 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.

[9] D. Nauck, F. Klawonn, and R. Kruse. *Neuronale Netze und Fuzzy Systeme*. Vieweg, 1994.

[10] 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.

[11] J. Stanley and E. Bate. *Neuronale Netze Computersimulation biologischer Intelligenz*. Systhema Verlag GmbH, 1991.

[12] 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.

[13] 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.

[14] G. Timmermann. Algorithmic music composition using artificial neural nets. `https://github.com/grant/algo-rhythm`. Abgerufen: 18.08.2016.

[15] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. *CoRR*, abs/1609.02612, 2016.