

# 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 darauffolgenden Kapitel werden die Grundlagen von NNs zusammengefasst. Um dann in Kapitel 2.3 vertiefend auf Faltungsnetze einzugehen.

### 2.1. Videoproduktion

Laura

### 2.2. Neuronale Netze

Laura

NNs finden unter anderem Anwendung bei der Steuerung von Robotern, Börsenkursanalysen, Medizin oder Fahrzeugsteuerung. In der Bildverarbeitung

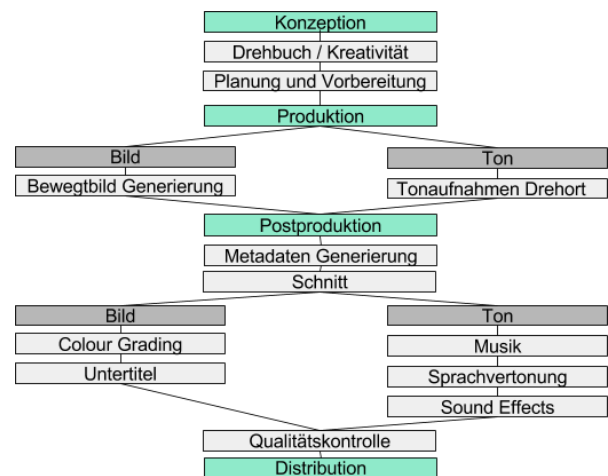
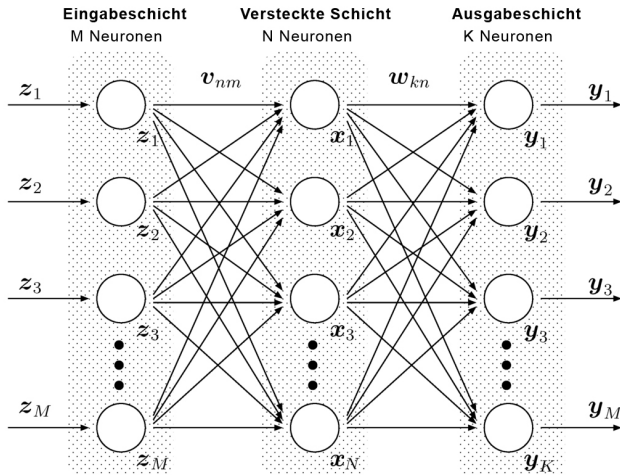


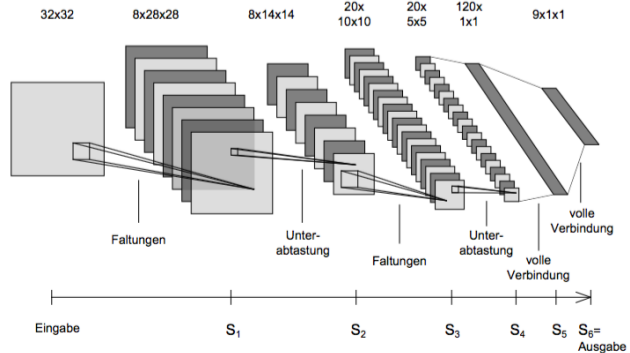
Abbildung 1. Grundlegende Arbeitsschritte einer Videoproduktion.



**Abbildung 2. Prinzipieller Aufbau MLP nach [3].**

werden NNs vor allem zur Klassifizierung genutzt. Sie sind vom menschlichen Gehirn inspiriert, welches laut [2] 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, wie in [2] und [6] beschrieben, in rück- und vorwärts gekoppelte Modelle unterteilt werden können.

Am häufigsten kommen sogenannte *Multilayer Perceptrons* (MLP) [1][4][6] 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 Eingabeschicht 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



**Abbildung 3. Prinzipieller Aufbau Faltungsnetz nach [5].**

und  $f$  für die Aktivierungsfunktion [6][2] 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.

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, 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.

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.

### 3. Vorverarbeitung

Vera

#### 3.1. Aktueller Stand

Vera

### 4. Produktion

Laura

#### 4.1. Aktueller Stand

Laura

#### 4.2. Szenendynamik nach Vondrick

Laura

### 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] S. Haykin. *Neural Networks A Comprehensive Foundation*. Macmillan College Publishing Company, 1994.
- [3] T. Isokawa, H. Nishimura, and N. Matsui. Quaternionic multilayer perceptron with local analyticity. *Information*, 3(4):756, 2012.
- [4] D. Nauck, F. Klawonn, and R. Kruse. *Neuronale Netze und Fuzzy Systeme*. Vieweg, 1994.
- [5] 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.
- [6] J. Stanley and E. Bate. *Neuronale Netze Computersimulation biologischer Intelligenz*. Systhema Verlag GmbH, 1991.