

Maschinelles Lernen - Informatik -

Rapp, DHBW Lörrach

09.02.2024

Inhaltsübersicht

1 Prüfungsaufgabe

2 Einführung

3 Convolution

Prüfungsaufgabe 5

Aufgabenteil 1: Messung der Ähnlichkeit von Künstlern

- a) Beschreiben Sie die im Programmcode `artists_similarity_cnn.py` beschriebene Vorgehensweise zur Messung der Ähnlichkeit von Künstlern.
- b) Clustern Sie Künstler in sinnvolle Gruppen.
Tip: Verwenden Sie hierfür die Projektion der Künstler in das Koordinatensystem aus dem Diagramm „Artists Projected on First Two PCs.“
- c) Beurteilen Sie die Aussagekraft der im Programmcode beschriebenen Ähnlichkeitsmessung von Künstlern.

Aufgabenteil 2: Generierung von Bildbeschreibungen

- Passen Sie die Parameter im Programm `Image_Caption_Generator_Transformer.py` so an, dass sinnvolle textbasierte Bildbeschreibungen (*engl.: image captions*) für Kunstwerke erzeugt werden.
Hinweis: Sinnvoll bedeutet insb., dass die Bildbeschreibungen potentiell für weitere Anwendungsfälle wie z.B. textbasiertes Clustern eingesetzt werden können.

Beispiele für Bewertungskriterien

- Aus dem Programmcode ist ersichtlich, dass die wesentlichen Konzepte aus der Vorlesung verstanden und auf die konkrete Problemstellung angewendet wurden.
- Der Fremdanteil am Programmcode ist mit entsprechenden Quellenangaben gekennzeichnet.
- ...

Präsentationsübung

Präsentationsübung am 23.02.2024

Präsentieren Sie während 2 Minuten Ihr **KI/ML-Lieblingsthema!**

Stellenausschreibungen für KI-Experten

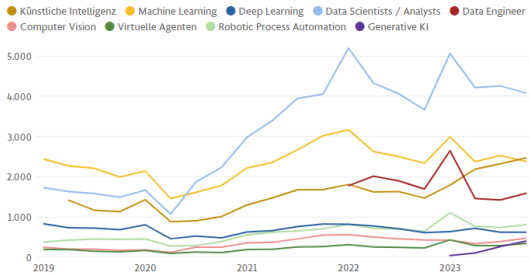
KI-JOBMONITOR

Fachleute für generative KI zunehmend gefragt

VON HOLGER SCHMIDT - AKTUALISIERT AM 31.01.2024 - 06:42

Stellenausschreibungen für KI-Experten

Ausgeschriebene Positionen in Deutschland für folgende Fachgebiete



Grafik: sacco. / Quelle: Index Gruppe 2024

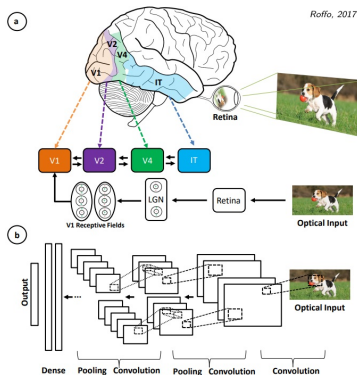
Unternehmen schichten ihre IT-Budgets in Richtung generativer KI um. Für Olaf Scholz geht das nicht schnell genug. Wissen über KI habe sich weder im Management noch in der Gesellschaft verbreitet.

Convolutional Neural Network

Das gefaltete künstliche neuronales Netz basiert auf einer Folge von mathematischen Faltungs- und Poolingoperationen und wird insb. in der maschinellen Bildverarbeitung eingesetzt.

Das Konzept dieses Netzwerktyps ist von **biologischen Prozessen** inspiriert, wie z.B. rezeptiven Feldern, welche unter anderem im visuellen Cortex zu finden sind.

Vergleich (a) Hirnareale des **primären visuellen Kortex** mit (b) Schichten eines **Convolutional Neural Networks (CNN)**



(a) Vier am ventralen visuellen Pfad beteiligte Brodmann Areale.

(b) AlexNet Convolutional Neural Network.

Paare aus Convolution und Pooling Schichten entsprechen "näherungsweise" der Hierarchie des biologischen visuellen Systems.

ImageNet Datenbank



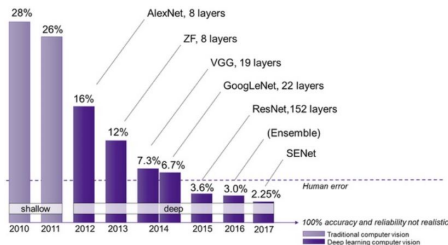
ImageNet: Große Datenbank mit 14 Millionen gelabelten und organisierten Bildern für Computer Vision.

ImageNet Datenbank Challenge

Computerbasiertes Sehen

Durch die Einführung neuer **Deep Learning Technologien** wurde computerbasiertes Sehen revolutioniert.

Ein wichtiger Meilenstein stellt die Einführung von **AlexNet** (*Alex Krizhevsky*) dar, einem frühen Convolutional Neural Network für die Bildverarbeitung:



ImageNet Large Scale Visual Recognition (Object detection+Classification) Challenge

Mit der Einführung von **ResNet** (*Microsoft Research, 2015*) wurde die menschliche Genauigkeit der Bilderkennung übertroffen. Sprungverbindungen reduzieren hier das Problem verschwindender Gradienten.

Anwendungen

Anwendungsbeispiele für Convolutional Neural Networks

• Bildklassifikation

- Gesichts- und Objekterkennung
- Autonomes Fahren (z.B. Klassifizierung von Verkehrszeichen)

• Spracherkennung

- Klassifizierung und Modellierung von Sätzen
- Maschinelles Übersetzen

Literaturempfehlung einschl. Link: [Kapitel 9 von Deep Learning von Goodfellow, Bengio und Courville](#)

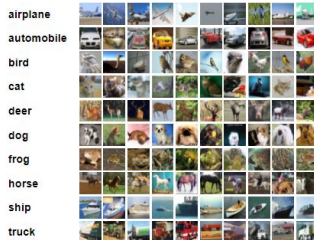
Bildklassifikation anhand von 2 Beispiel-Datensätzen

MNIST-Datenbank



Beispiele (Modified National Instit. of Standards and Techn.)

CIFAR-10



Beispiele (Canadian Institute For Advanced Research)

- Datenbank mit Bildern von handschriebenen Ziffern
- jede Ziffer ist als 28x28 Pixel großes Graustufen-Bild gespeichert
- gelabelte Trainings- (60.000 Beispiele) und Testdaten (10.000)

- Datenbank mit Bildern
- 32x32 RGB-Fotos
- gelabelte (10 Klassen) Trainings- (50.000 Beispiele) und Testdaten (10.000)

Convolution

Convolution mit geometrischer Interpretation

Die **Convolution** (=Faltung) liefert für zwei reelle Funktionen f und g eine dritte Funktion $f * g$:

$$h(t) = (f * g)(t) := \int_{\tau=-\infty}^{\tau=\infty} f(\tau)g(t - \tau)d\tau$$

h : **Feature Map** (Output), f **Input**, $*$ **Convolution Operator**,
 g **Kern**

Geometrische Interpretation

- Anschaulich bedeutet die Faltung $f * g$, dass jeder Wert von f durch das mit g **gewichtete Mittel** der ihn umgebenden Werte ersetzt wird.

Genauer wird für den Mittelwert $(f * g)(t)$ der Funktionswert $f(\tau)$ mit $g(t - \tau)$ gewichtet.

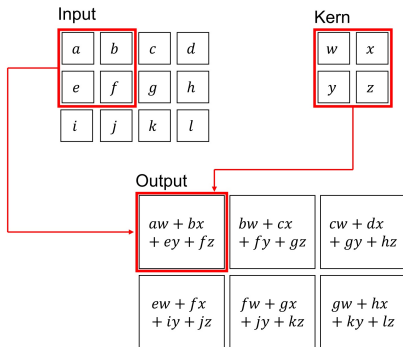
- Es findet eine „*Verschmierung von f* “ statt, d.h. f wird durch **gespiegelte** und **verschobene** Versionen von g überlagert.

Convolution in der Praxis

In Anwendungen **mehrdimensionale Arrays** als **Input** und **Kern**

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n)$$

Beispiel



$$I \in \mathbb{K}^{m \times n}, K \in \mathbb{K}^{k \times k}$$

$$\Rightarrow K * I \in \mathbb{K}^{(m-k+1) \times (n-k+1)}$$

$$I \in \mathbb{K}^{3 \times 4}, K \in \mathbb{K}^{2 \times 2}$$

$$\Rightarrow K * I \in \mathbb{K}^{2 \times 3}$$

Rechenbeispiel

Input I

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

Kern K

1	0	-1
1	0	-1
1	0	-1

 $\xrightarrow{K * I}$
Output S

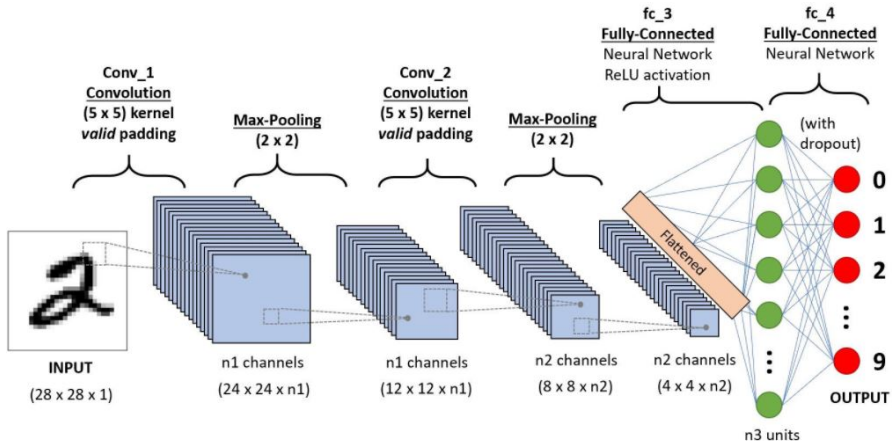
0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

$$S(1,1) = \sum_{m=-1}^1 \sum_{n=-1}^1 I(1+m, 1+n)K(m,n)$$

$$= 10 + 10 + 10 + 0 + 0 + 0 - 10 - 10 - 10 = 0$$

⇒ Kern hat Filterfunktion

Schematischer Aufbau eines CNN



CNN Sequenz für die Klassifikation handgeschriebener Zahlen

Aus: *medium.com, Kundathil*

CNN Code-Beispiel in Python

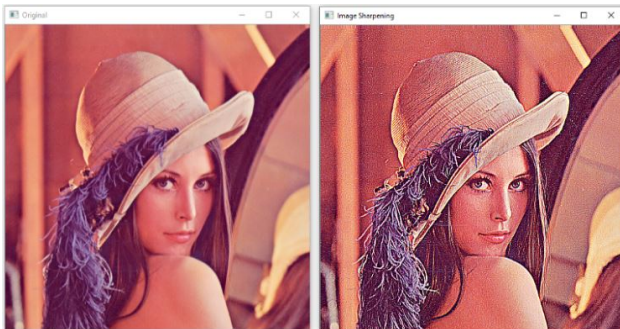
Einfaches **CNN Modell** für das Training von MNIST Daten

```
from keras.models import Sequential
from keras.layers import Conv2D, Dense, MaxPool2D, Flatten
model = Sequential()
## Convolution Layer
model.add(Conv2D(filters=32, kernel_size=(4,4), input_shape=(28,28,1), activation="relu"))
model.add(MaxPool2D(pool_size=(2,2)))
## Converting from 2D -> 1D
model.add(Flatten())
## Dense Layer
model.add(Dense(128, activation="relu"))
model.add(Dense(10, activation="softmax"))
## Compiling The Model
model.compile(loss="categorical_crossentropy", optimizer="rmsprop", metrics=["accuracy"])
```

Aus: medium.com, Kundathil

Beispiel Schärfungsfilter

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

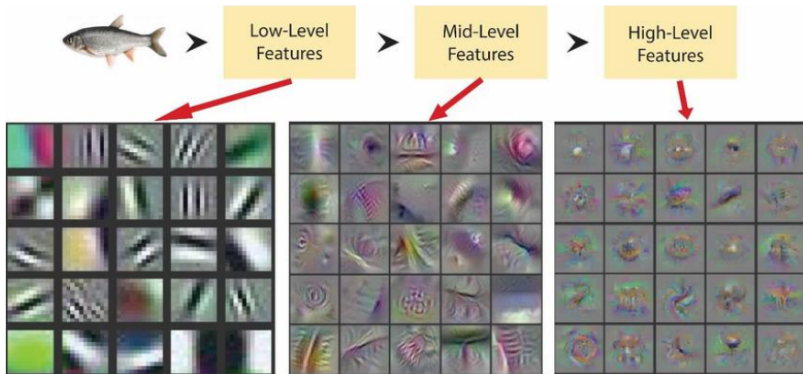


Aus: <https://medium.com/@bdhuma/6-basic-things-to-know-about-convolution-daef5e1bc411>

Ursprüngliches Bild (links) sowie nach der Anwendung des Schärfungsfilters der Größe 3x3 (rechts)

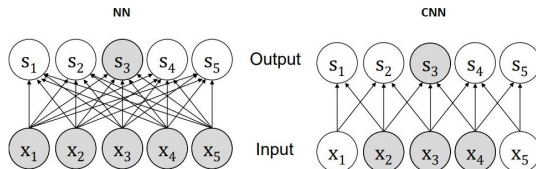
Visualisierungsbeispiele

Visualisierungsbeispiele der Feature Maps im Convolutional Neural Network



Die erste Schicht stellt einfache Muster wie z.B. Kanten und Steigungen dar. Die höheren Schichten abstraktere Features.

Vergleich: Fully Connected NN vs. CNN



Fully Connected Neural Network

- Jedes Neuron eines Layers ist mit allen Neuronen des nachfolgenden Layers verbunden
- Jeder Output interagiert mit jeder Input Einheit
- Gewicht beschreibt Interaktion zwischen Output und Input

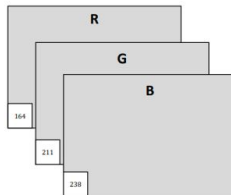
Convolutional Neural Network

- Kern kleiner als Input
- Somit weniger Operationen zur Berechnung des Outputs (*Sparse Interactions*)
- Ein Set Gewichte lernen statt ein separates für jede Stelle
→ „Gewichte Sharing“
- Reduziert Speicherbedarf und verbessert die Effizienz ▶

Erweiterung auf Farbbilder

- Bisher: Input Bild der Größe Höhenpixel \times Breitenpixel
- Damit können nur Schwarz-Weiß Bilder dargestellt werden
- Buntes Bild hat pro Pixel 3 Werte

R	G	B	Farb-Beispiel
255	165	000	
164	211	238	
154	205	050	



→ Größe Input = Höhe \times Breite \times 3

→ Kern hat Dimension $k \times k \times 3$

Fazit - Convolutional Neural Network

- State-Of-The-Art in Bild- und Audioverarbeitung
- Fehlerraten: Teilweise besser als Mensch
- Mit Grafikprozessoren lassen sich CNNs sehr effizient trainieren
- Vorteile CNN gegenüber Fully Connected Neural Networks
 - Robustheit gegenüber Störungen, die in den Testdaten vorhanden sind
 - Weniger Speicherplatzbedarf
 - Einfacheres und besseres Training