

7. Übungsblatt

Aufgabe 33 Convolution

In der Bildverarbeitung werden Graustufenbilder oft als zweiparametrische Funktion $f(x, y)$ dargestellt und als Matrix beschrieben. Dabei sind x und y die Koordinaten eines Bildpunktes und der Funktionswert / Zellwert gibt die Helligkeit an.

Das folgende Bild ist in Graustufen gegeben:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 10 & 10 & 10 & 0 \\ 0 & 10 & 10 & 10 & 10 & 0 \\ 0 & 10 & 10 & 0 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Convolutional Neural Networks sind in der Lage effizient Bilddaten unabhängig von Rotation und Position enthaltener Objekte zu verarbeiten. Hierfür werden Kernel verwendet, welche jeweils mit Ausschnitten der Matrix verrechnet werden. Die auf diese Weise berechneten Convolved Features können zur Erkennung von Objekten verwendet werden.

Der Sobel-Operator dient der Kantenerkennung in Bilddaten. Nimmt man an, dass bei einer Kante die Helligkeit stark wechselt, so kann man Kanten an den Extremwerten der ersten Ableitung der Funktion f erkennen. Gegeben seien die Operatoren S_x und S_y .

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- a) Berechnen Sie die Features G_x und G_y durch ausschnittsweise Multiplikation der Operatoren S_x und S_y mit der Matrix A . Verwenden Sie Zero Padding von 2.

$$G_x = S_x * A$$

$$G_y = S_y * A$$

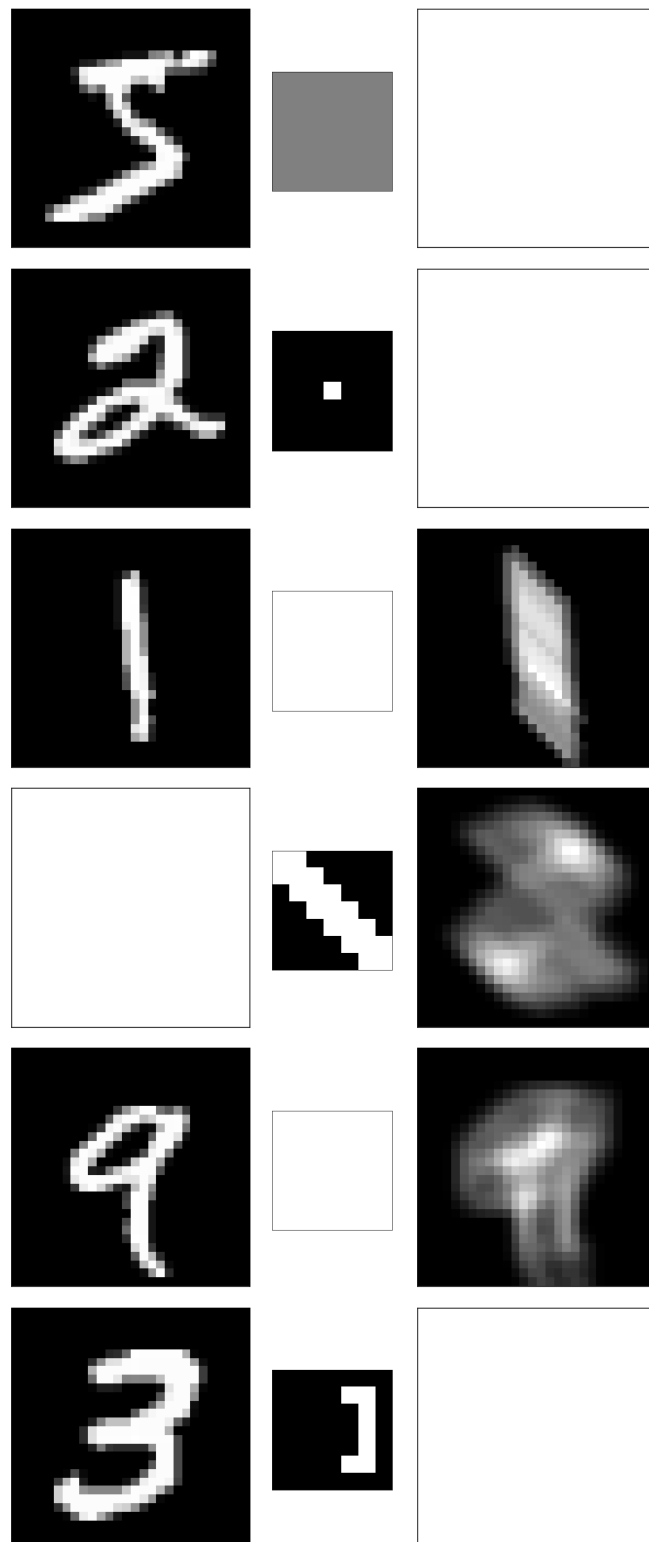
- b) Aus den richtungsabhängigen Matrizen soll jetzt eine richtungsunabhängige Matrix G erstellt werden. Dafür werden **die Einträge** von G_x und G_y **jeweils** quadriert, dann aufsummiert und aus dem Ergebnis die Wurzel gezogen.

$$G = \sqrt{G_x^2 + G_y^2}$$

Berechnen sie die Matrix G mit Hilfe der Teilergebnisse aus Aufgabenteil a) und beschreiben sie das Endergebnis!

Aufgabe 34 Convolution

Skizzieren Sie das fehlende Bild. Hierbei befindet sich in der linken Spalte das Ausgangsbild, in der Mitte der Kern und in der rechten Spalte die resultierende Feature Map.



Aufgabe 35 Receptive Field

Nehmen Sie an, dass ein CNN Bilder mit 32x32 Pixeln als Eingabe erhält. Bestimmen Sie, aus wie vielen Schichten von Konvolution mit einem Stride von 1 und einem 2x2 Kernel ein Netz bestehen muss, damit das ganze Bild von der letzten Schicht einbezogen werden kann.

Was bedeutet dies für das Klassifizieren von hochauflösenden Bildern ?

Aufgabe 36 Stride, Padding, Pooling

Die Ausgabedimension O eines beliebigen Convolution Layers lässt sich mit Hilfe der Formel

$$O = \frac{(W - F + 2P)}{S} + 1$$

berechnen, wobei W die Eingabedimension, F die Filtergröße, P das Zero Padding und S der Stride ist.

- Berechnen Sie die Ausgabedimension O für eine Eingabe der Dimension 32×32 , Filtern der Dimension 6×6 , Zero Padding von 2 und Stride von 2.
Wie ändert sich die Ausgabe, wenn wir die Filter auf 5×5 und den Stride auf 3 setzen?
- Wie viele Reihen muss man durch Zero Padding an eine Eingabe beliebiger Größe anfügen, damit Ausgabe und Eingabe die selbe Dimension besitzen?
- Wenden Sie Max pooling und Average pooling mit einem Filter der Dimension 2×2 und Stride 2 auf die folgende Matrix an.
Warum werden Pooling Layer in CNNs eingesetzt? Welche Vorteile bieten sie? Warum wird meist nur Max pooling verwendet?

$$A = \begin{bmatrix} 1 & 2 & 2 & 7 & 4 & 9 \\ 2 & 3 & 3 & 6 & 1 & 8 \\ 4 & 9 & 2 & 9 & 6 & 11 \\ 4 & 7 & 6 & 11 & 6 & 7 \\ 2 & 3 & 7 & 2 & 8 & 1 \\ 1 & 2 & 6 & 3 & 4 & 9 \end{bmatrix}$$

Aufgabe 37 Bonus: Automatische Differenzierungsframework (5)

Mit dieser Aufgabe wollen wir uns Klassifikation zuwenden. Dafür implementieren wir die Kreuzentropie, die auf eine Softmaxaktivierung folgt als eine differenzierbare Funktion. Dafür sollten Sie wie folgt vorgehen:

- Implementieren Sie für die Tensorklasse die exp Funktion
- Implementieren Sie für die Tensorklasse den natürlichen Logarithmus
- Erweitern Sie `sum(dim=None)` um den Parameter `dim` der es erlaubt entlang einer bestimmten Dimension die Elemente aufzuaddieren. Wenn `dim=None` soll die Funktion sich wie bisher verhalten.

Bsp.:

$$\text{sum}\left(\begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \end{pmatrix}, \text{dim} = 1\right) = \begin{pmatrix} 6 \\ 12 \end{pmatrix}$$

$$\text{sum}\left(\begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \end{pmatrix}, \text{dim} = 0\right) = (4 \ 6 \ 8)$$

$$\text{sum}\left(\begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \end{pmatrix}, \text{dim} = \text{None}\right) = (18)$$

- Implementieren Sie crossentropy. Erinnern Sie sich dafür an den besprochenen Logarithmus der Softmaxfunktion.

Als nächstes wollen wir das erste Mal ein Klassifikationsproblem mit unserem Framework lösen. Dafür gibt es wieder eine Vorlage. Diesmal arbeiten wir mit dem Iris Datensatz <http://archive.ics.uci.edu/ml/datasets/iris>. Das Laden des Datensatzes im gleichen Ordner, so die Vorbereitung der Daten zu batches befindet sich in der Vorlage. Implementieren Sie nun folgende Schritte:

- Implementieren Sie ein Model.
- Trainieren Sie das Model.
- Erreichen Sie ein Testaccuracy von mindestens 90%.