Künstliche Intelligenz WS 20/21
Aufgaben für die 10. Woche
18. bis 24. Januar 2021

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Aufgabe 10.1

Auf dem letzten Übungsblatt haben Sie gesehen wie man Feature selbst berechnet. Eine von vielen Ideen bei neuronalen Netzen ist, dass man sich die Feature nicht mehr selbst überlegen muss, sondern man lässt das Netz selbst Feature entwickeln.

In den nächsten zwei Wochen beschäftigen wir uns mit der Bildverarbeitung. In diesem Zusammenhang werden die Begriffe *Cross-Correlation* (Kreuzkorrelation) und *Convolution* (Faltung) eine zentrale Rolle einnehmen. Mit diesen Verfahren werden wir in der Lage sein sogenannte *Feature Maps* zu berechnen. Ein ähnliches Verfahren haben Sie bereits in der zweiten Woche auf eindimensionalen Sensordaten kennengelernt. Zunächst definieren wir eine neue Indizierung für einen Filter F, wie folgt:

$$F := \begin{pmatrix} f(-\frac{h-1}{2},-\frac{w-1}{2}) & \cdots & f(-\frac{h-1}{2},-1) & f(-\frac{h-1}{2},0) & f(-\frac{h-1}{2},1) & \cdots & f(-\frac{h-1}{2},\frac{w-1}{2}) \\ \vdots & \ddots & \vdots & \vdots & \vdots & \iddots & \vdots \\ f(-1,-\frac{w-1}{2}) & \cdots & f(-1,-1) & f(-1,0) & f(-1,1) & \cdots & f(-1,\frac{w-1}{2}) \\ f(0,-\frac{w-1}{2}) & \cdots & f(0,-1) & f(0,0) & f(0,1) & \cdots & f(0,\frac{w-1}{2}) \\ f(1,-\frac{w-1}{2}) & \cdots & f(1,-1) & f(1,0) & f(1,1) & \cdots & f(1,\frac{w-1}{2}) \\ \vdots & \iddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f(\frac{h-1}{2},-\frac{w-1}{2}) & \cdots & f(\frac{h-1}{2},-1) & f(\frac{h-1}{2},0) & f(\frac{h-1}{2},1) & \cdots & f(\frac{h-1}{2},\frac{w-1}{2}) \end{pmatrix} \tag{10.1}$$

aus Gründen der Symmetrie und Eindeutigkeit, wählt man $h, w \in \mathbb{N}$ mit $h, w \geq 3$ meist ungerade. Für ein Eingabebild I und einen Kernel K ist die **Cross-Correlation** wie folgt definiert:

$$g(x,y) := \sum_{i=-\frac{k-1}{2}}^{\frac{k-1}{2}} \sum_{j=-\frac{k-1}{2}}^{\frac{k-1}{2}} I(x+i, y+j) \cdot f(i,j) \tag{10.2}$$

Im folgenden betrachten wir ein Beispiel mit Eingabebild I der Größe 4×5 und einen Filter F der Größe 3×3:

$$I := \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \qquad F := \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

$$\begin{aligned} G = I * F &= \begin{pmatrix} (1\cdot1+1\cdot0+1\cdot1+0\cdot0+1\cdot1+1\cdot0+0\cdot1+0\cdot0+1\cdot1) & (1\cdot1+1\cdot0+0\cdot1+1\cdot0+1\cdot1+1\cdot0+0\cdot1+1\cdot0+1\cdot1) \\ (0\cdot1+1\cdot0+1\cdot1+0\cdot0+0\cdot1+1\cdot0+0\cdot1+1\cdot0+1\cdot1) & (1\cdot1+1\cdot0+1\cdot1+0\cdot0+1\cdot1+1\cdot0+0\cdot1+1\cdot0+0\cdot1) \end{pmatrix} \\ &= \begin{pmatrix} 4 & 3 \\ 2 & 4 \end{pmatrix} \end{aligned}$$

Die Matrix G nennen wir **Feature-Map**. Manche Machine-Learning Libraries verwenden statt der

Cross-Correlation eine Convolution. Die diskrete **Convolution** ist wie folgt definiert:

$$g(x,y) \coloneqq \sum_{i=-\frac{k-1}{2}}^{\frac{k-1}{2}} \sum_{j=-\frac{k-1}{2}}^{\frac{k-1}{2}} I(x-i, y-j) \cdot f(i,j) \tag{10.3}$$

Der Unterschied zwischen einer Cross-Correlation und einer Convolution ist, dass der Filter bei der Convolution zuvor um 180° rotiert wird. Oft spricht man hier auch von *flippen*, da eine 180° Drehung äquivalent zu einer horizontalen Spiegelung gefolgt von einer vertikalen Spiegelung ist (oder umgekehrt). Da der Filter im obigen Beispiel symmetrisch ist, ändert sich das Ergebnis für eine Convolution nicht.
Im Zusammenhang von Cross-Correlation und Convolution hört man öfter die Begriffe *Stride* und *Padding*. Dabei bezeichnet *Stride* um wie viel Zeilen oder Spalten, der Filter verschoben wird (im obigen Beispiel wäre $Stride = 1$). Wie Sie vielleicht bemerkt haben, reduziert sich die Ausgabegröße je nach Filtergröße und Stride. Um diesem Problem Abhilfe zu schaffen, erweitert man das Eingabebild um den Rand; Entweder mit Nullen oder den Bildwerten. Diese Erweiterung nennt sich Padding. Da wir dies im obigen Beispiel nicht gemacht wurde, gilt dort $Padding = 0$.
Lösen Sie die folgenden Aufgaben:

(a) Überlegen Sie sich eine Formel, wie man von der Eingabegröße, Filtergröße, Stride und Padding auf die Ausgabegröße schließen kann.

(b) Verwenden Sie die Formel in (a) um die Funktion `cross_correlation` im Jupyter Notebook zu vervollständigen. Um zu testen ob ihre Implementierung das macht was sie soll, wenden wir die Cross-Correlation Funktion von OpenCV auf das Eingabebild an.