Übungsblatt 5: Convolutions-Layer und Batchnormalisierung
BA-INF 153: Einführung in Deep Learning für Visual Computing

Deadline:	05.06.2024 - 14:00 via eCampus	
Tutoren:	Alina Pollehn	s6aapoll@uni-bonn.de
	Johannes van de Locht	s6jovand@uni-bonn.de
Übungsgruppenleitung:	Jan Müller	muellerj@cs.uni-bonn.de

Theoretische Aufgaben (*15 Punkte*)

a) "Receptive Field" und Outputgröße *(5 Punkte)* Das "Receptive Field" eines Filters in einem CNN ist der Bereich im Eingabebild, der das Ergebnis des Filters beeinflusst. Ein Filter eines 3×3-Konvolution-Layers, der direkt auf die Eingabe angewendet wird, hat beispielsweise ein "Receptive Field" von 3×3. Wenn wir eine weitere 3×3-Konvolution auf dieses Ergebnis anwenden, haben die Filter der zweiten Schicht ein "Receptive Field" von 5×5. Ähnlich verhält es sich bei einer Pooling-Operation mit einer Filtergröße von 2, die auf die Eingabe angewendet wird: Das "Receptive Field" jedes gepoolten Wertes beträgt 2×2. Zur Berechnung der Receptive Field Größe müssen wir also nicht zwischen Konvolution und Pooling Operationen unterscheiden sondern müssen lediglich die Filtergröße und den Stride Wert berücksichtigen.

1. *(2 Punkte)* Geben Sie eine Formel an, mit der die **Größe des "Receptive Field"** nach Anwendung von n Kernel-Operationen mit quadratischen Filtern der Größe $k_i \times k_i$ und einem Stride-Wert s_i berechnet werden kann. Berechnen Sie mit Hilfe Ihrer Formel die "Receptive Field" Größe für das Konvolutions-Netzwerk:

$$\mathrm{Conv}_{k_1=7,s_1=1} \to \mathrm{Conv}_{k_2=5,s_2=1} \to \mathrm{Conv}_{k_4=3,s_4=1} \to \mathrm{MaxPool}_{k_3=2,s_3=2}$$

   Hinweis: Sie können annehmen, dass das Eingabebild ausreichend groß ist. Es müssen keine "edge-cases" betrachtet werden.

2. *(3 Punkte)* Geben Sie eine allgemeine Formel an, mit der die **Größe des Output** nach Anwendung einer Kernel-Operationen mit quadratischen Filtern der Größe $k_i \times k_i$, einem Stride-Wert s_i und einem Padding mit p_i Pixeln berechnet werden kann. Die Paddinggröße p_i gibt an wie viele Pixel an den Rändern des Bildes hinzugefügt werden und hat einen Einfluss auf welche Randpixel die Kernel-Operation angewendet werden kann. Berechnen Sie mit Hilfe Ihrer Formel die Outputgröße eines RGB-Bildes $3 \times 224 \times 224$ nach der Anwendung des Konvolutions-Netzwerks

$$\mathrm{Conv}_{\mathrm{in}_1=3,\mathrm{out}_1=16,k_1=7,s_1=2,p_1=0} \to \mathrm{Conv}_{\mathrm{out}_2=16,\mathrm{out}_2=32,k_2=5,s_2=1,p_2=0} \to$$
$$\mathrm{Conv}_{\mathrm{in}_3=32,\mathrm{out}_3=32,k_3=3,s_3=1,p=1} \to \mathrm{MaxPool}_{k_4=2,s_4=2,p_4=0}.$$

Hinweis: Bei Max-Pooling wird die Operation auf jeden Kanal der Eingabe separat angewendet. Daher ist die Anzahl der Eingabe und Ausgabe Kanäle identisch.

b) Batchnormalisierung *(4 Punkte)* In der Vorlesung wird Batch-Normalisierung definiert als

$$H' = \frac{H - \mu}{\sigma} \text{ wobei } \mu = \frac{1}{m}\sum_{i=1}^{m} H_{i,:} \text{ und } \sigma = \sqrt{\sigma + \frac{1}{m}\sum_{i=1}^{m}(H_{i,:} - \mu)_i^2}.$$

- ***(2 Punkte)*** Die Definition aus der Vorlesung beschreibt die Berechnung von Batch-Normalisierung bei Daten die als Vektoren dargestellt werden. Wie wird Batch-Norm. berechnet wenn die Daten 2-dim. dargestellt werden (etwa die Ausgabe eines Konv.-Layer)?
- ***(2 Punkte)*** Es gibt Anwendungen in denen Batch-Normalisierung nicht verwendet wird. Recherchieren Sie nach einem solchen Anwendungen, begründen sie warum Batch-Norm dort nicht angewendet werden kann und finden Sie heraus welchen Technik stattdessen verwendet wird.

c) Gradient eines Convolutionslayers *(6 Punkte)* Betrachten Sie die folgende Situation: Gegeben sei ein quadratisches Eingangsbild $x \in \mathbb{R}^{n\times n}$ und ein Konvolutiongewichte $w \in \mathbb{R}^{m\times m}$. Sei m ungerade und seien die Indizes von w so verschoben, dass $w_{0,0}$ der Wert in der Mitte von w ist. Dann sei die diskrete (Kreuz-)Konvolution $o = w * x$ von w und x definiert als

$$o_{i,j} = \sum_{k=-m}^{m}\sum_{l=-m}^{m} w_{k,l} \cdot x_{i-k,j-l} \text{ for } i, j = 1, ..., n.$$

Eine Max-Pooling-Operation p auf x mit Downsampling-Faktor d (Filtergröße d und Stride d) kann definiert werden als

$$p_d(x)_{i,j} = \max\left(x_{\pi_{ij}(1)}, ..., x_{\pi_{ij}(d^2)}\right)$$

wobei $\pi_{ij}(k)$ eine Funktion ist, die die entsprechenden Indizes aus der Eingabe angibt, die zusammengeführt werden. Bei einem 4×4-Eingabebild x und $d = 2$ ergibt sich zum Beispiel

$$\begin{bmatrix} 1,1 & 1,2 & 1,3 & 1,4 \\ 2,1 & 2,2 & 2,3 & 2,4 \\ 3,1 & 3,2 & 3,3 & 3,4 \\ 4,1 & 4,2 & 4,3 & 4,4 \end{bmatrix} = \begin{bmatrix} \pi_{1,1}(1) & \pi_{1,1}(2) & \pi_{2,1}(1) & \pi_{2,1}(2) \\ \pi_{1,1}(3) & \pi_{1,1}(4) & \pi_{2,1}(3) & \pi_{2,1}(4) \\ \pi_{1,2}(1) & \pi_{1,2}(2) & \pi_{2,2}(1) & \pi_{2,2}(2) \\ \pi_{1,2}(3) & \pi_{1,2}(4) & \pi_{2,2}(3) & \pi_{2,2}(4) \end{bmatrix}$$

und ein daraus resultierendes gepooltes Bild

$$o' := p_2(x) = \begin{bmatrix} p_2(x)_{1,1} & p_2(x)_{1,2} \\ p_2(x)_{2,1} & p_2(x)_{2,2} \end{bmatrix}.$$

Ein MSE-Fehler E in dieser zweidimensionalen Umgebung ist der Mittelwert aller n_o^2 quadrierten Differenzen, wobei n_o die Größe der Ausgangsdimension ist, d.h. $o, y \in \mathbb{R}^{n_o \times n_o}$ und

$$E(o, y) = \frac{1}{n_o^2} \sum_{i,j=1}^{n_o} (o_{i,j} - y_{i,j})^2.$$

1. ***(3 Punkte)*** Berechnen Sie die partielle Ableitung von $E(o, y)$ für die Konvolution $o = w * x$ nach den Gewichten $w_{s,t}$ der Konvolution.
2. ***(3 Punkte)*** Berechnen Sie die partielle Ableitung von $E(o', y')$ für das 2-Max-Pooling $o' = p_2(x)$ nach den Eingaben $x_{i,j}$.