Convolution neural networks - Understanding layer input sizes/shapes

I am unsure about how input shape of each hidden convolutional layers should be.
Here is an example:
My input is an image 8 x 8 and my first convolution layer has a \textit{window} size of 3x3 should the next layers input size look like 6x6? assuming no zero padding.
Then if I do maxpooling of size 2x2 right after the first convolution the next layer should look like 3 x 3?
I was told that at the end of the network I should have a 1x1 pixel (hyper pixel) but I am unsure why we should have this.

all 2 comments □ subscribe

[-] NasenSpray 2 points 20 hours ago
My input is an image 8 x 8 and my first convolution layer has a \textit{window} kernel size of 3x3 should the next layers input size look like 6x6? assuming no zero padding.
Yup, \(H \times W\) conv on \(N \times M\) input \(\Rightarrow (N-H+1) \times (M-W+1)\) output
Then if I do maxpooling of size 2x2 right after the first convolution the next layer should look like 3 x 3?
Only if its non-overlapping, i.e., with stride 2 (aka \(2 \times 2 / s2\))
I was told that at the end of the network I should have a 1x1 pixel (hyper pixel) but I am unsure why we should have this.
I don't know how to put it into better words, but just do what makes the most sense. There isn't really an 'one-size-fits-all' approach.

[-] klop2031 [S] 1 point 20 hours ago
Thanks, I needed the clarification

\[
\text{Input layer} \quad \xrightarrow{\text{Conv1}} \quad \xrightarrow{\text{Conv2}} \quad \xrightarrow{\text{fc1}} \quad \xrightarrow{\text{fc2}} \quad \text{output layer} \xrightarrow{\pi} \text{prediction}
\]

\[
P(\text{pad}) = 0 \quad \rightarrow \quad \frac{(W-F+2P)}{S} + 1
\]
\[
S (\text{stride}) = 1 \quad \rightarrow \quad 8 - 3 + 1 = 6
\]
\[
(8 - 3 + 1) \times (8 - 3 + 1) = 6 \times 6
\]