Deep Learning for Computer Vision Ex3

Guy Lutsker

Q:

HW3- Theory

A 1-D stride-2-convolution layer is applied twice sequentially to an input 1-D tensor, with the same filter denoted as k_2. The input is padded with zeros, only once before the sequence of two convolution layers is applied, so that the final output size is exactly 1/4 of the input size. Find an equivalent filter k_4 that when applied just once with a stride-4 convolution layer and the same padding, yields the exact same result.

Guidance: Recall that convolution layer (with appropriate padding) actually performs cross-correlation- discrete non-circular convolution with the **flipped** filter. Also note that a convolution with stride is equivalent to a convolution followed by subsampling. Thus, k_4 is defined by:

$$x * \tilde{k_4} \downarrow_4 = (x * \tilde{k_2} \downarrow_2) * \tilde{k_2} \downarrow_2$$

where $\tilde{k}$ is flipped k and $\downarrow_s$ means subsampling by a factor of s (e.g., subsampling by 2 means taking every second value).

Hint: use the notion of dilated convolution.

Note: Perfect score will be obtained by a full formal proof. However, correct final answer with a reasonable explanation will be considered too.

Solution:

Let us first denote $n \in \mathbb{N}$ our signal length, and our signal as $x \in \mathbb{R}^n$. We shall also denote our kernel as $k_2 \in \mathbb{R}^c$ $c \in \mathbb{N}$. From the guidance we know we should use the definition of $\tilde{k}$ to describe convolution as per the definition.

Let us look at:

$$\left(x \star \widetilde{k_2}\right)(u) = \sum_{m=\infty}^{\infty} x(m) \cdot \widetilde{k_2}(u-m)$$

In order for us to maintain the size of the input we pad the image with zeroes. The size of the padding in general is half the size of the kernel - $\frac{c}{2}$. And so:

$$= \sum_{m=-\frac{c}{2}}^{n+\frac{c}{2}} x(m) \cdot \widetilde{k_2}(u-m)$$

Now, recall that our operator also uses subsampling by a factor of 2, denoted by the operator $\downarrow_2$.

And so Let us look at:

$$\sum_{m=-\frac{c}{2}}^{n+\frac{c}{2}} x(m)\cdot\widetilde{k_2}(u-m)\downarrow_2 = \sum_{m\in\left\{\left[-\frac{c}{2},\ n+\frac{c}{2}\right] mod_2\right\}} x(m)\cdot\widetilde{k_2}(u-m)$$

Now let's look at our final expression we were actually asked to compute:

$$\left(x\star\widetilde{k_2}\downarrow_2\star\widetilde{k_2}\downarrow_2\right)(u) =$$

$$\sum_{m=-\frac{c}{2}}^{n+\frac{c}{2}}\left[\sum_{t\in\left\{\left[-\frac{c}{2},\ n+\frac{c}{2}\right] mo\ \ _2\right\}} x(t)\cdot\widetilde{k_2}(u-t)\right]\cdot\widetilde{k_2}(u-m)\downarrow_2$$

Recall that convolution is symmetric such that $f\star g = g\star f$, and so we can write:

$$= \sum_{m\in\left\{\left[-\frac{c}{2},\ n+\frac{c}{2}\right] mo\ \ _2\right\}}\widetilde{k_2}(m)\cdot\left[\sum_{t\in\left\{\left[-\frac{c}{2},\ n+\frac{c}{2}\right] mod_2\right\}} x(t)\cdot\widetilde{k_2}(u-t-m)\right]$$

We are almost at the notation of convolution, all that's missing is that we change the order of the terms. Luckly we can change the order of summation here (discrete Fubini?):

$$= \sum_{t\in\left\{\left[-\frac{c}{2},\ n+\frac{c}{2}\right] mod_2\right\}} x(t)\cdot\left[\sum_{m\in\left\{\left[-\frac{c}{2},\ n+\frac{c}{2}\right] mod_2\right\}}\widetilde{k_2}(m)\cdot\widetilde{k_2}(u-t-m)\right]$$

Now, notice that this is in fact in convolution notation, and that since both summations go over even indices, we get that the signal x is reduced by a scale of 4 – equivalent of using the $\downarrow_4$.

And so, we can write:

$$\widetilde{k_4}(u) = \left[\sum_{m\in\left\{\left[-\frac{c}{2},\ n+\frac{c}{2}\right] mod_2\right\}}\widetilde{k_2}(m)\cdot\widetilde{k_2}(u-t-m)\right]\ \forall u\ s.t\ u_{mod_4} = 0$$

And finally:

$$\left(x\star\widetilde{k_2}\downarrow_2\star\widetilde{k_2}\downarrow_2\right)(u) = (x\star\widetilde{k_4}\downarrow_4)(u)$$