Header styling inspired by CS 70: https://www.eecs70.org/

Collaborators

I worked with the following people on this assignment:

- Teja Nivarthi: 3036508567
- Nikhil Maserang: 3036978230

Problem 1

Convolution, especially in 2D, is considered to be one of the core operations of image and signal processing, particularly within the field of deep learning and computer vision. By enabling the extraction of features from images and signals, convolutions enable tasks such as object detection and noise reduction. However, one of the biggest constraints of the 2D convolution is often the computational buden due to the sheer volume of parameters required – proportional to the channel and the kernel size.

The concept of separability of 2D signal can also be applied in coonvolution, and it offers a promising solution to mitigate these challenges. By decomposing a 2D convolution into two successive 1D convolutions, separable convolutions drastically reduce the number of parameters and the computational expense.

Let's try to examine how separability works in convolution!

a) Recall that the definition of Convolution 2D is

$$y[m,n] = x[m,n] * h[m,n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i,j] \cdot h[m-k, n-j]$$

Determine if y is separable assuming that the 2D Kernel/Filter h is separable.

Solution: Assuming that h is separable, it means that we can write $h[m,n] = h_1[m]h_2[n]$. Then, we use the fact that convolution is commutative, so instead of writing $x * h$, we write out $h * x$:

$$h[m,n] * x[m,n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} h_1[i]h_2[j]x[m-i, n-j]$$

Then, we can separate this out too:

$$\sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} h_1[i]h_2[j]x[m-i,n-j] = \sum_{i=-\infty}^{\infty} h_1[i] \underbrace{\left(\sum_{j=-\infty}^{\infty} h_2[j]x[m-i,n-j] \right)}_{x * h_2}$$

note that there was nothing special about choosing to separate out h_2 here either, so the same can be done with h_1. Hence, we can write:

$$y[m,n] = h_1[m] * (h_2[n] * x[m,n]) = h_2[n] * (h_1[m] * x[m,n])$$

I think the moral is then that because h is separable, then it means that we can convolve along the m and n separately, then combine them at the end. □

---

b) Now let's try some examples. Compute $y[1,1]$ by using separable convolution method and compare the reuslt with the output of normal convolution 2D.

$$\begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

where the second matrix represents the filter/kernel H.

i) Use normal 2D Convolution:

Solution: Here the normal 2D convolution just tells us to flip the second matrix (which does nothing to this matrix, ironically), and then just stack them together to find the element value:

$$y[1,1] = 10(1) + 20(2) + \cdots + 80(2) + 90(1) = 800$$

□

---

ii) Use a separable 2D Convolution (Hint: express the kernel as a vector-vector multiplication):

Solution: We have to separate out the second matrix into a vertical $h_1[m]$ and a horizointal $h_2[n]$. It turns out that (from the solutions lol)

$$\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} = \begin{bmatrix}1\\ 2\\ 1\end{bmatrix} \times \begin{bmatrix}1 & 2 & 1\end{bmatrix}$$

so now that we have our two matrices, and to find $y[1,1]$, all we need to do is convolve by each one individually:

$$\begin{bmatrix}10 & 20 & 30\\ 40 & 50 & 60\\ 70 & 80 & 90\end{bmatrix} * \begin{bmatrix}1\\ 2\\ 1\end{bmatrix} = \begin{bmatrix}160\\ 200\\ 260\end{bmatrix}$$

Then, we have to convolve by the row matrix now:

$$\begin{bmatrix}160 & 200 & 240\end{bmatrix} * \begin{bmatrix}1 & 2 & 1\end{bmatrix} = 800$$

which matches the earlier result. □

---

c) The benefit of separable convolution might not be clear in part (b) as we did separable convolution for only 1 sample; in this case, more steps of multiplications are needed than regular 2D convolution does.

However, when it comes to the full separable convolution for all 9 input data at once, we can find why the separable convolution can be powerful.

Compute the full 2D convolution using separable convolution:

Solution: So we'll just do the same thing as the previous part, but now do it for the entire matrix. The trick is to go vertically: we take the dot product of each row, starting with $\begin{bmatrix}0 & 10 & 40\end{bmatrix}$ with $\begin{bmatrix}1 & 2 & 1\end{bmatrix}$ and end with $\begin{bmatrix}60 & 90 & 0\end{bmatrix}$. Therefore, the resulting matrix is:

$$\begin{bmatrix}10 & 20 & 30\\ 40 & 50 & 60\\ 70 & 80 & 90\end{bmatrix} * \begin{bmatrix}1\\ 2\\ 1\end{bmatrix} = \begin{bmatrix}60 & 90 & 120\\ 160 & 200 & 240\\ 180 & 210 & 240\end{bmatrix}$$

Now, doing the same thing but vertically, we get:

$$\begin{bmatrix}60 & 90 & 120\\ 160 & 200 & 240\\ 180 & 210 & 240\end{bmatrix} * \begin{bmatrix}1 & 2 & 1\end{bmatrix} = \begin{bmatrix}210 & 360 & 330\\ 520 & 800 & 680\\ 570 & 840 & 690\end{bmatrix}$$

□

---

d) Now, assume that you are trying to use an ideal low-pass filter, which is a rect function. Let's examine if this ifltering can take advantage of the separability. Calculate the two-dimensional Fourier transform of a rectangle of unit height (z-direction) and size $a \times b$ centered about the origin.

$$f(x,y) = \begin{cases}1 & \text{if } |x| < a/2 \text{ and } |y| < b/2\\ 0 & \text{else}\end{cases}$$

Solution: I looked at the provided solutions, and I know that they go through the whole derivation. However, I find it far easier to argue that since we know that $f(x,y)$ is separable, then we can just treat it as a 1D rect function along x and y. Then, since we've shown many times that the FT of a rect function is a sinc, then all we need to do is combine them togehter:

$$F(\omega_x, \omega_y) = ab\,\mathrm{sinc}(a\omega_x)\,\mathrm{sinc}(b\omega_y)$$

□

Problem 2

An image described by the function

$$x(t_1, t_2) = 2\cos(3t_1 + 4t_2)$$

is sampled at $T_1 = T_2 = 0.4\pi$.

a) Find the 2DFT of $x(t_1, t_2)$. In other words, find $X(\omega_1, \omega_2)$.

Solution: This problem was entirely done in review session. Here, we use Euler's formula twice, and remember that:

$$\cos(\omega_0 t) \longleftrightarrow 2\pi(\delta(\omega + \omega_0) + \delta(\omega - \omega_0))$$

Therefore, we have:

$$X(\omega_1, \omega_2) = (2\pi)^2[\delta(\omega_1 - 3, \omega_2 - 4) + \delta(\omega_1 + 3, \omega_2 + 4)$$

□

---

b) Is this signal bandlimited? Show mathematically. If bandlimited, find the bandwidth.

Solution: This signal is clearly bandlimited, with $|\omega_1| < 3$ and $|\omega_2| < 4$. So, $\omega_{M_1} = 3$ and $\omega_{M_2} = 4$. □

---

c) Does the current sampling rate obey the Nyquist Theorem? Explain.

Solution: The current sampling has $\omega_{s_1} = \frac{2\pi}{0.4\pi} = 5 = \omega_{s_2}$. Since this value is not larger than twice the bandwidth for either ω_{M_1} or ω_{M_2}, this does not obey the Nyquist theorem. □

---

d) Find the expression for the reconstructed image $\tilde{x}(t_1, t_2)$. Keep in mind that we also need a round of filtering with an ideal low-pass filter! Does the reconstructed image match with the original image?

Solution: The sampled signal looks like:

$$X_s(\omega_1, \omega_2) = \frac{25}{4} \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \delta(\omega_1 - 3 - 5k_1, \omega_2 - 4 - 5k_2) + \delta(\omega_1 + 3 - 5k_1, \omega_2 + 4 - 5k_2)$$

Then, we pass this through an ideal low pass filter, which is of the form:

$$H(\omega_1, \omega_2) = \begin{cases} T_1 T_2 & |\omega_1| < \frac{\omega_{s_1}}{2} \text{ and } |\omega_2| < \frac{\omega_{s_2}}{2} \\ 0 & \text{else} \end{cases}$$

Therefore, after sampling, we have the signal:

$$\tilde{X}(\omega_1, \omega_2) = (2\pi)^2[\delta(\omega_1 - 2, \omega_2 - 1, \omega_1 + 2, \omega_2 + 1)$$

now doing the inverse Fourier transform, we get:

$$\tilde{x}(t_1, t_2) = 2\cos(2t_1 + t_2)$$

the reconstructed image doesn't match the original image. This makes sense though, since we had already mentioned earlier that it fails the Nyquist theorem. □
