

Imagens Biomédicas – Lab03-Pyton

Tutores: André Arruda / Maíra Suzuka Kudo / Eric Rocha Santos

Professor: Matheus Cardoso Moraes

CORRELAÇÃO, CONVOLUÇÃO E KERNEL (MÁSCARA)

EXERCÍCIOS:

1. → No Editor, fazer dois vetores, kernel w e sinal f , com os valores e comprimentos descrito abaixo:

a. $w = [1\ 2\ 3\ 2\ 8];$

b. $f = [0\ 0\ 0\ 1\ 0\ 0\ 0\ 0];$

Crie um vetor *padding*, vetor f acima preenchido com zeros. Para o preenchimento obtenha um vetor **pad** com comprimento de $L_w - 1$, e insira-o nas posições iniciais e finais de f , para que f possua comprimento de:

- $L = L_o + 2 \cdot (L_w - 1)$
- nos quais L_o é o comprimento inicial do sinal e L_w é o comprimento do kernel.
- $f_{padding} = [\underline{0\ 0\ 0\ 0}\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ \underline{0\ 0\ 0\ 0}];$

Obs.: Use a função `np.zeros` e `np.concatenate((pad,f,pad), axis=None)`

2. No Editor, fazer um loop de “for” que faça a varredura de todo o vetor *fpadding*.

Obs.: Pode usar `L = np.shape(fpadding)`, lembrando que o valor do comprimento está em `L[0]`

3. Altere o loop de “for” para fazer a operação de correlação cruzada entre *w* e *fpadding*, ou seja a soma dos produtos entre os vetores.

```
for...  
    cor[i] = ....
```

Obs.: declare `cor = np.zeros(.....)`

para multiplicar elemento de vetores faça `a[0:5]*b[0:5]`

use a função `np.sum` para somar

4. Use a função de abaixo para verificar o resultado.

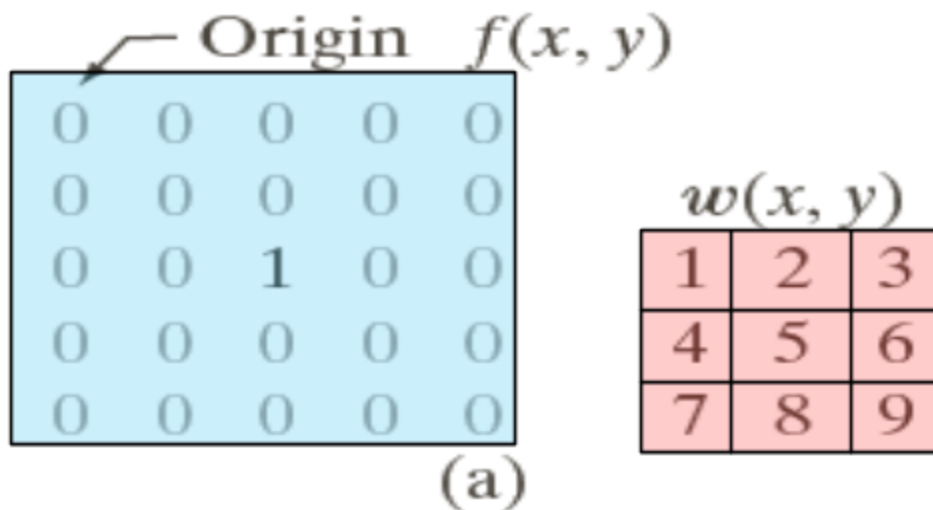
```
corFuncao = np.correlate(f,w,"full")
```

5. Recorte o início e o fim do vetor do resultado Final *cor*, para que este tenha o tamanho do sinal original de *f* ou seja, 8 posições.

```
ccrop[0:7] = cor[?:?:?]
```

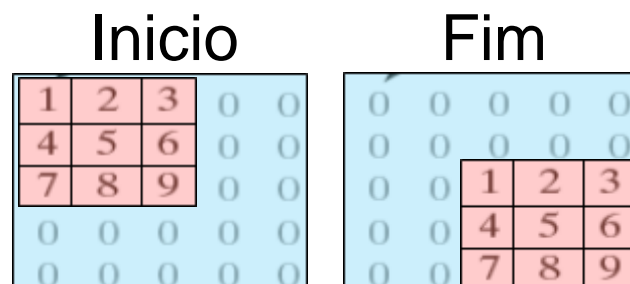
6. Usando a função “zeros”, crie uma matriz chamada “f” e outra “c” de 5 linhas e 5 colunas preenchidas com zeros, na matriz “f”, faça *f* tipo *float*. Substitua o valor na posição [2,2] pelo valor 1 (Figura). Adicionalmente, crie uma máscara *w* com os valores abaixo. Usando a função “*np.shape*”, armazene o número de linhas *M*, e colunas *N* de *f*.

Obs.: criar Matrizes exemplo `a = np.array([[2,3],[4,5]])`



7. Faça uma função de correlação 2D entre a matriz f e a máscara w , usando apenas 2 “for”, apenas para varrer a imagem. Para não precisar fazer o preenchimento com zeros, **ajuste o início e o fim do loop** de tal forma que a máscara esteja posicionada dentro da matriz f , como mostrado na figura abaixo.

```
for...Varrer linhas da imagem
  for... Varrer colunas da imagem
    cor2[l,c] = .....
```



8. (**Deixar por último como desafio**) Faça novamente função de correlação 2D entre a matriz f e o máscara w , *como anteriormente. Porém agora usando 4 "for", 2 para varrer a imagem e 2 para varrer a máscara fazendo multiplicação e soma.*

```
for...Varrer linhas da imagem
    for... Varrer colunas da imagem

        soma..

        for...Varrer linhas da mascara
            for... Varrer colunas da mascara

                .....

            end

        end

    end
    c(x,y) = .....
end
end
```

9. Novamente faça a correlação 2D entre a matriz f e o máscara w , usando a função abaixo. *O resultado foi o mesmo? Por quê o tamanho da matriz de correlação não é o mesmo?*

```
corFuncao2 = scipy.signal.correlate2d(f, w, boundary='symm', mode='same')
```

Obs.: import scipy.signal

10. Leia e já normalize como float a imagem e exiba a imagem mamograph.pgm.

```
i0 = cv2.imread(' mamograph.pgm', 0) # Gray
in0 = skimage.img_as_float(i0)
```

- a. Crie uma máscara w de dimensões 3x3 preenchidas por “1s”, use a função “ones”. Multiplique por um “*fator*” para que cada coeficiente de correlação computado represente a média dos pixels dentro da área da máscara. Qual deve ser o valor desse fator para que isto ocorra?

$$w = \text{fator} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- b. Faça a correlação entre $in0$ e w , usando a função “scipy.signal.correlate2d”. Exiba os coeficientes de correlação como uma imagem. O que houve com a imagem? Por quê?

```
in0Filt = scipy.signal.correlate2d (...);  
in0Filt = skimage.exposure.rescale_intensity(in0Filt, in_range=(0,1))  
plt.figure()  
plt.title('imFilt0')  
plt.imshow(in0Filt, cmap='gray')
```

- c. Refaça a correlação para máscaras de 5x5 e 10x10, recalculando o fator para cada uma das matrizes. Quais os novos valores? Exiba as duas e explique o que houve com as imagens?