Filtragem Linear

Alexandre Xavier Falcão

Instituto de Computação - UNICAMP

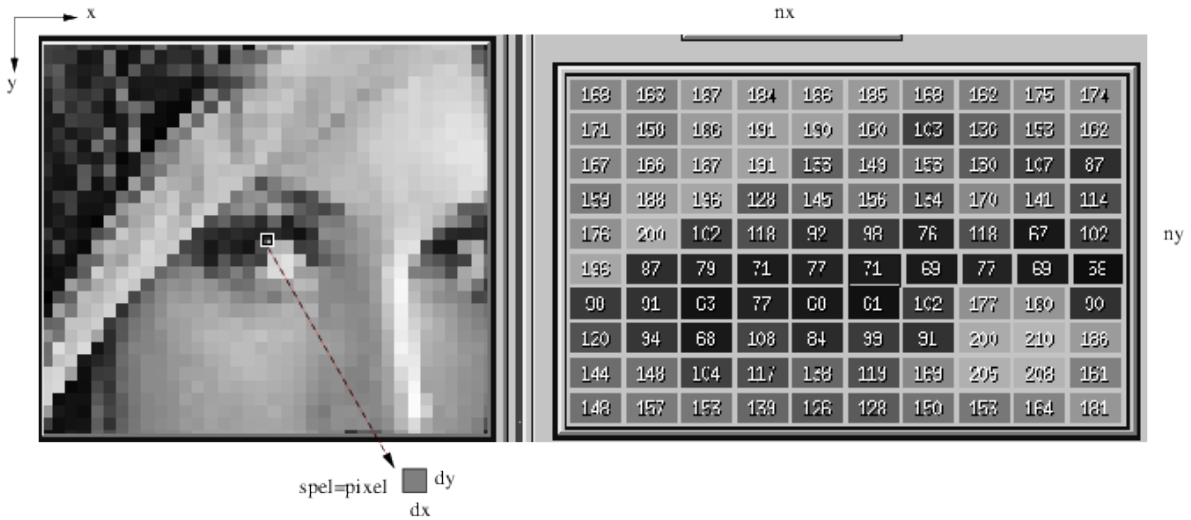
afalcao@ic.unicamp.br

Introdução

- Filtros lineares têm hoje um papel fundamental na extração de medidas (características) em redes neurais convolucionais.
- A filtragem linear resulta da convolução entre uma imagem (sinal) e um *kernel*.
- Portanto, para entender a filtragem linear, precisamos dos conceitos de imagem, adjacência, kernel, e convolução.

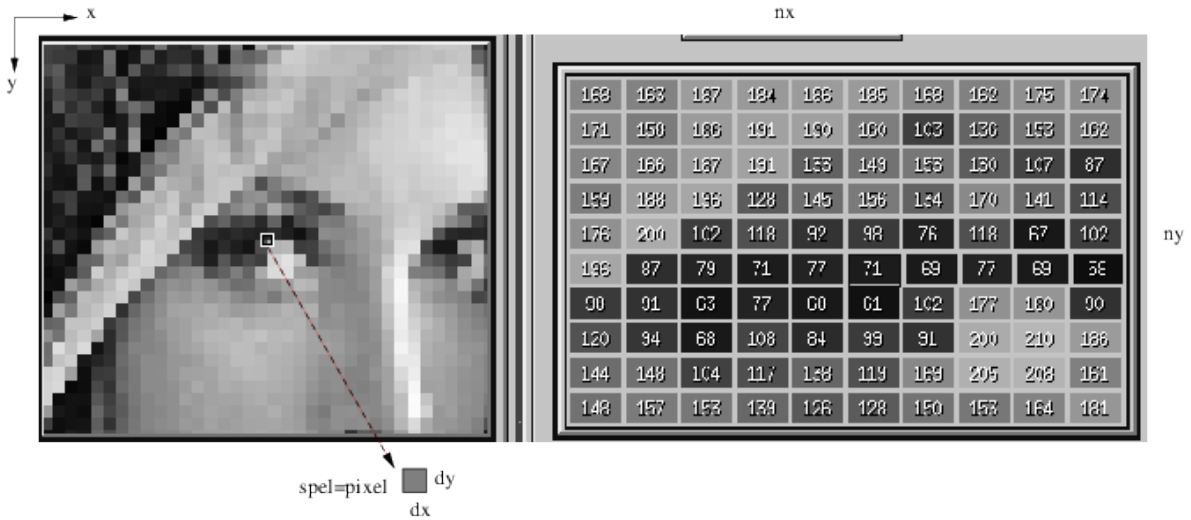
Imagem

Uma imagem $\hat{I}$ é um par $(D_I, \mathbf{I})$ em que a função $\mathbf{I}$ pode associar um conjunto de medidas $\mathbf{I}(p) = \{I_1(p), I_2(p), \ldots, I_n(p)\}$ para cada elemento (pixel) do seu domínio espacial $D_I \in Z^m$. No caso mais simples, $m = 2$, $p = (x_p, y_p)$, $|D_I| = n_x \times n_y$, $n = 1$, e $\mathbf{I}(p)$ é um escalar — i.e., a imagem é monocromática.



Imagem

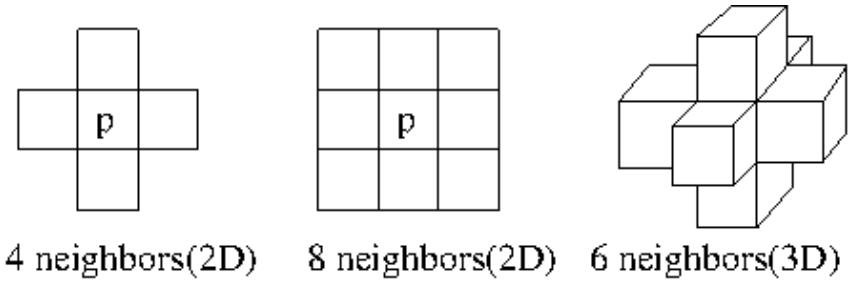
Uma imagem $\hat{I}$ é um par $(D_I, \mathbf{I})$ em que a função $\mathbf{I}$ pode associar um conjunto de medidas $\mathbf{I}(p) = \{I_1(p), I_2(p), \ldots, I_n(p)\}$ para cada elemento (pixel) do seu domínio espacial $D_I \in Z^m$. No caso mais simples, $m = 2$, $p = (x_p, y_p)$, $|D_I| = n_x \times n_y$, $n = 1$, e $\mathbf{I}(p)$ é um escalar — i.e., a imagem é monocromática.

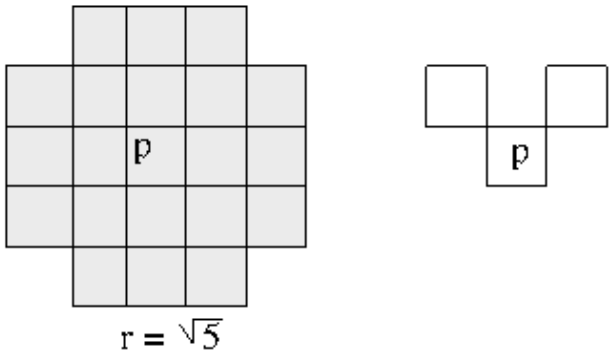


Neste caso, simplificando a notação, a imagem é o par $\hat{I} = (D_I, I)$.

Relação de Adjacência

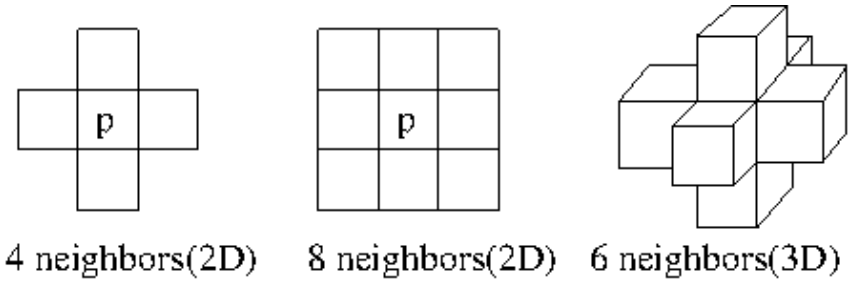
Uma relação de adjacência é um relação binária definida em $D_I \times D_I$ com base em uma métrica entre pixels.

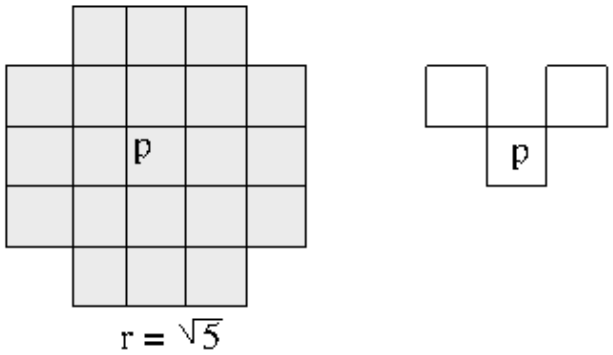




Relação de Adjacência

Uma relação de adjacência é um relação binária definida em $D_I \times D_I$ com base em uma métrica entre pixels.





Para simplificar, vamos considerar apenas relações $\mathcal{A}$ simétricas derivadas de

$$\mathcal{A}: \{(p,q) \in \mathcal{A} | \|q-p\| \leq r\},$$

onde $r > 0$ e $\|.\|$ é a norma Euclideana.

Detalhes de implementação

- A imagem pode ser armazenada em um vetor $I[p]$, $p = 0, 1, \ldots, |D_I| - 1$, tal que $x_p = p\%n_x$ e $y_p = p/n_x$. Note que existe uma relação direta entre o índice p do vetor e o pixel $p = (x_p, y_p)$.

Detalhes de implementação

- A imagem pode ser armazenada em um vetor $I[p]$, $p = 0, 1, \ldots, |D_I| - 1$, tal que $x_p = p\%n_x$ e $y_p = p/n_x$. Note que existe uma relação direta entre o índice p do vetor e o pixel $p = (x_p, y_p)$.

- Para um dado valor de $r > 0$, vamos considerar $q_i = (x_{q_i}, y_{q_i}) \in D_I$, $i = 0, 1, 2, \ldots, |\mathcal{A}| - 1$, como os pixels adjacentes de $p = (x_p, y_p) \in D_I$ (i.e., $(p, q_i) \in \mathcal{A}$).

Detalhes de implementação

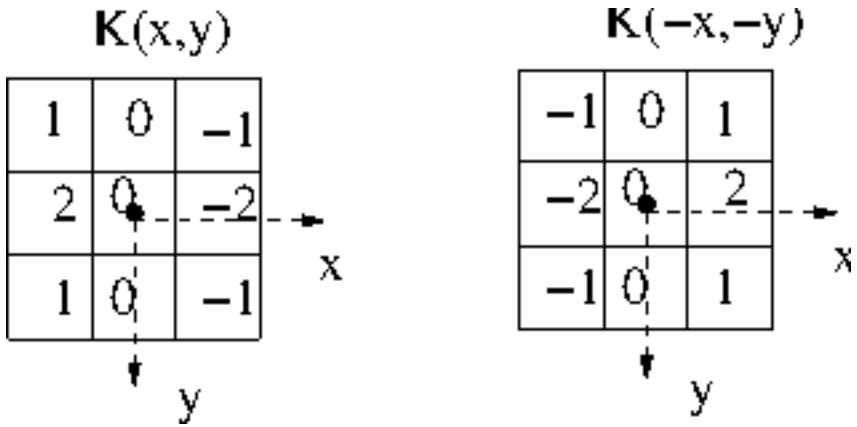
- A imagem pode ser armazenada em um vetor $I[p]$, $p = 0, 1, \ldots, |D_I| - 1$, tal que $x_p = p\%n_x$ e $y_p = p/n_x$. Note que existe uma relação direta entre o índice p do vetor e o pixel $p = (x_p, y_p)$.

- Para um dado valor de $r > 0$, vamos considerar $q_i = (x_{q_i}, y_{q_i}) \in D_I$, $i = 0, 1, 2, \ldots, |\mathcal{A}| - 1$, como os pixels adjacentes de $p = (x_p, y_p) \in D_I$ (i.e., $(p, q_i) \in \mathcal{A}$).

- Então podemos armazenar apenas os deslocamentos $(dx_i, dy_i) = (x_{q_i}, y_{q_i}) - (x_p, y_p)$ em vetores $dx[i]$ e $dy[i]$ e acessar todo pixel q_i a partir de qualquer pixel p por $x_{q_i} = x_p + dx[i]$ e $y_{q_i} = y_p + dy[i]$, $i = 0, 1, 2, \ldots, |\mathcal{A}| - 1$. Normalmente $dx[0] = dy[0] = 0$, facilitando a inclusão/exclusão do pixel p como seu adjacente.

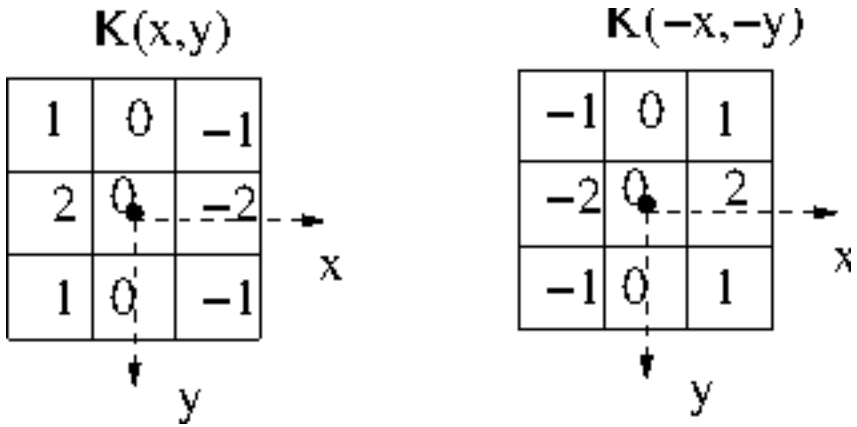
Kernel

- Um kernel $\hat{K}$ é um par $(\mathcal{A}, K)$ que associa um peso fixo $K(q-p) = w_i$, $i = 0, 1, 2, \ldots, |\mathcal{A}| - 1$, para cada adjacente q_i de p.



Kernel

- Um kernel $\hat{K}$ é um par $(\mathcal{A}, K)$ que associa um peso fixo $K(q-p) = w_i$, $i = 0, 1, 2, \ldots, |\mathcal{A}| - 1$, para cada adjacente q_i de p.



- Então basta representar um kernel pelos vetores $dx[i]$, $dy[i]$, e $w[i]$.

Convolução

- A filtragem linear é essencialmente o resultado da convolução de uma imagem $\hat{I} = (D_I, I)$ por um kernel $\hat{K} = (\mathcal{A}, K)$ ("imagem móvel").

- A rigor, o kernel precisa ser refletido com relação à origem de $\mathcal{A}$, mas podemos assumir que o kernel já está refletido sem perda de generalidade.

- A convolução $\hat{I} * \hat{K}$ entre $\hat{I}$ e $\hat{K}$ resulta, portanto, uma imagem $\hat{J} = (D_J, J)$ onde, para todo $p \in D_J$,

$$J(p) = \sum_{\forall (p,q) \in \mathcal{A}} I(q)K(q-p) = \sum_{i=0}^{|\mathcal{A}|-1} I(q_i)w_i.$$

  A rigor, $D_I \subset D_J$, mas constumamos adotar $D_J = D_I$ na filtragem linear.

Algoritmo de filtragem linear

Entrada: $\hat{I} = (D_I, I)$ e $\hat{K} = (\mathcal{A}, K)$.
Saída: $\hat{J} = (D_I, J)$.

1. Para todo $p \in D_J$, faça
2. $\quad J(p) \leftarrow 0$.
3. $\quad$ Para todo $(p, q) \in \mathcal{A}$, tal que $q \in D_I$, faça
4. $\quad\quad J(p) \leftarrow J(p) + I(q)K(q - p)$.

Algoritmo de filtragem linear

Entrada: $\hat{I} = (D_I, I)$ e $\hat{K} = (\mathcal{A}, K)$.
Saída: $\hat{J} = (D_I, J)$.

1. Para todo $p \in D_J$, faça
2. $\quad J(p) \leftarrow 0$.
3. $\quad$ Para todo $(p, q) \in \mathcal{A}$, tal que $q \in D_I$, faça
4. $\quad\quad J(p) \leftarrow J(p) + I(q)K(q - p)$.

Isto é, basta varrer os vetores de deslocamento $dx[i]$, $dy[i]$, calcular q_i, verificar se $q_i \in D_I$, e então acumular no vetor $J[p]$ o valor $I[q_i]w[i]$.

Exercício

Um exercício interessante é resolver a convolução por multiplicação matricial. Basta criar uma matriz onde as colunas armazenam os valores $I(p)$ dos pixels de $\hat{I}$ e as linhas armazenam os valores $I(q_i)$ de seus adjacentes. O kernel neste caso é uma matriz com uma única linha, onde os elementos das colunas são os pesos w_i. Multiplica-se a matriz do kernel pela matriz da imagem estendida pela adjacência.

Exemplo de filtragem linear

O realce de bordas usando os kernels de Sobel é um exemplo típico.

$$K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Sendo $\hat{G}_x = \hat{I} * \hat{K}_x$ e $\hat{G}_y = \hat{I} * \hat{K}_y$, $\vec{G}(p) = G_x(p)\vec{i} + G_y(p)\vec{j}$ é dito vetor gradiente em p, o qual ponta para a direção de maior variação de brilho.

Exemplo de filtragem linear

O realce de bordas usando os kernels de Sobel é um exemplo típico.

$$K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

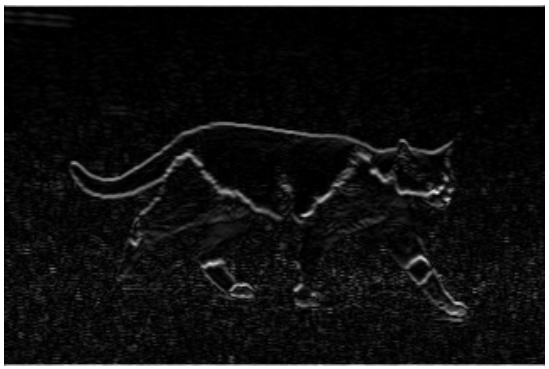
Sendo $\hat{G}_x = \hat{I} * \hat{K}_x$ e $\hat{G}_y = \hat{I} * \hat{K}_y$, $\vec{G}(p) = G_x(p)\vec{i} + G_y(p)\vec{j}$ é dito vetor gradiente em p, o qual ponta para a direção de maior variação de brilho.

Exemplo de filtragem linear

O realce de bordas usando os kernels de Sobel é um exemplo típico.

$$K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Sendo $\hat{G}_x = \hat{I} * \hat{K}_x$ e $\hat{G}_y = \hat{I} * \hat{K}_y$, $\vec{G}(p) = G_x(p)\vec{i} + G_y(p)\vec{j}$ é dito vetor gradiente em p, o qual ponta para a direção de maior variação de brilho.



Exemplo de filtragem linear

O realce de bordas usando os kernels de Sobel é um exemplo típico.

$$K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Sendo $\hat{G}_x = \hat{I} * \hat{K}_x$ e $\hat{G}_y = \hat{I} * \hat{K}_y$, $\vec{G}(p) = G_x(p)\vec{i} + G_y(p)\vec{j}$ é dito vetor gradiente em p, o qual ponta para a direção de maior variação de brilho.