Homework: Image denoising

Realized by: Eya Ghamgui

Exercice 1.1 :

* we have the preceptron function :

$$f(x,y) = \begin{cases} 1 & \text{if } w_1 x + w_2 y + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

* we have the NAND gate :  $\overline{AB}$

A	B	out
0	0	1
0	1	1
1	0	1
1	1	0

$$\Rightarrow \text{NAND}(x,y) = \begin{cases} 1 & \text{if } x \neq 1 \text{ or } y \neq 1 \\ 0 & \text{if } x = 1 \text{ and } y = 1 \end{cases}$$

- if $x = 1$ and $y = 1$: we want that :
  $f(1,1) = 0 \Rightarrow w_1 + w_2 + b \leq 0$

Let's take as a example : $\begin{cases} w_1 = -1 \\ w_2 = -1 \\ b = 2 \end{cases}$

- if $x=0$ and $y=1$: $-1 \times 0 - 1 \times 1 + 2 = 1 > 0$
- if $x=0$ and $y=0$: $-1 \times 0 - 1 \times 0 + 2 = 2 > 0$
- if $x=1$ and $y=0$: $-1 \times 1 - 1 \times 0 + 2 = 1 > 0$

For these three cases, $f(x,y) = 1$.

Thus, if $x \neq 1$ or $y \neq 1 \Rightarrow f(x,y) = 1$

if $x = 1$ and $y = 1 \Rightarrow f(x,y) = 0$

$\Rightarrow$ we can say that by choosing $\{ w_1 = -1 ;\ w_2 = -1 ;\ b = 2 \}$, we obtain $f(x,y) = \text{NAND}(x,y)$

Exercice n 1.2 :

* We have a Patch-wise DCT transform can be written as:

$$\begin{cases} k = 0, \ldots, N-1 \\ Y_k = 2\alpha_k \sum_{j=0}^{N-1} X_j \cos\left(\pi\left(j + \frac{1}{2}\right)\frac{k}{N}\right) ; \quad \alpha_k = \begin{cases} \frac{1}{\sqrt{4N}} & k=0 \\ \frac{1}{\sqrt{2N}} & k=1,\ldots,N-1 \end{cases} \end{cases}$$

* We have a convolution formulation:

$$y[k] = x(k) * k(k) = \sum_{n=0}^{N-1} x(n)\, k(k-n)$$

* In this case, we have 4×4 patches $\Rightarrow N = 4$.

$$\Rightarrow Y_k = 2\,\alpha_k \sum_{j=0}^{N-1} X_j \cos\left(\pi \left(j + \tfrac{1}{2}\right) k/N\right)$$

$$= \sum_{j=0}^{N-1} X_j \times \left(2\,\alpha_k \cos\left(\pi \left(j+\tfrac{1}{2}\right) \tfrac{k}{N}\right)\right)$$

$$= \sum_{j=0}^{N-1} X_j \times \left(2\,\alpha_k \cos\left(\pi\left(k - (k-j) + \tfrac{1}{2}\right) \tfrac{k}{N}\right)\right)$$

$$= \sum_{j=0}^{N-1} X_j \times Z_k(k-j) = X * Z_k$$

where : $Z_k(j) = 2\,\alpha_k \cos\left(\pi\left(k - j + \tfrac{1}{2}\right)\tfrac{k}{N}\right) \;\forall\; 0 \leq j \leq N-1$

this is the kth kernel of convolution.

We have 4 kernels and each one has 4 elements.

$\Rightarrow$ we have shown that the patch-wise DCT transform (4×4 patches) of grayscale image can be implemented and represented using convolutions.

Exercice 1.3:

We have $f(x) = g(wx + b)$ with g is a sigmoid function and w is a (4×3 matrix).

$$\Rightarrow \begin{cases} g(x) = \dfrac{1}{1 + e^{-x}} \\ y(x) = wx + b \end{cases} \Rightarrow f(x) = g(y(x))$$

* $\dfrac{\partial f(x)}{\partial x} \overset{\text{chain Rule}}{=} \dfrac{\partial f(x)}{\partial y} \times \dfrac{\partial y}{\partial x}$

$$= \frac{\partial g(y(x))}{\partial y} \times \frac{\partial y(x)}{\partial x} = \frac{-e^{-y(x)}}{\left(1 + e^{-y(x)}\right)^2} \times \frac{\partial y(x)}{\partial x}$$

$\longrightarrow y(x + h) - y(x) = w(x + h) - w(x)$

$= wh = \langle w^T, h\rangle \Rightarrow \dfrac{\partial y(x)}{\partial x} = w^T$

$$\frac{\partial f(x)}{\partial x} = \frac{-e^{-y(x)}}{1 + 2e^{-y(x)} + e^{-2y(x)}}\, w^T = \frac{-1}{2 + e^{y(x)} + e^{-y(x)}}\, w^T$$

$$\Rightarrow \circledast\ \frac{\partial f(x)}{\partial x} = \frac{-1}{2 + e^{Wx+b} + e^{-(Wx+b)}} W^T$$

$$\circledast\ \frac{\partial f(x)}{\partial b} = \frac{\partial g(y(x))}{\partial b}$$

(chain Rule) $$= \frac{\partial g(y(x))}{\partial y} \times \frac{\partial y}{\partial b} = \frac{-1}{2 + e^{Wx+b} + e^{-(Wx+b)}} \mathbb{1}$$

$$\star\ \frac{\partial f(x)}{\partial W} \underset{\text{(chain Rule)}}{=} \frac{\partial g(y(x))}{\partial y} \times \frac{\partial y}{\partial W} = \frac{-1}{2 + e^{Wx+b} + e^{-(Wx+b)}} \times \frac{\partial y}{\partial W}$$

We have : $Wx + b = \begin{pmatrix} w_1 x \\ w_2 x \\ \vdots \\ w_n x \end{pmatrix} + b$ with w_i are the lines of matrix W.

or : $(w_i + h)x - w_i x = hx = \langle x^T, h \rangle$

$$\Rightarrow \frac{\partial y}{\partial w_i} = \frac{\partial (Wx + b)}{\partial w_i} = \begin{pmatrix} 0 \\ x^T \\ 0 \\ 0 \end{pmatrix} \leftarrow i^{th} \text{ position.}$$

$$\Rightarrow \circledast\ \frac{\partial f(x)}{\partial W} = \frac{-1}{2 + e^{Wx+b} + e^{-(Wx+b)}} \begin{pmatrix} x^T \\ \vdots \\ x^T \end{pmatrix}$$

Exercice 1.4 :

$f_i(x, \theta_i)$: network layer with θ_i and input x.

Multi layer network: $\begin{cases} F(x) = f_3(y; \theta_3) \text{ with } y = f_2(f_1(x,\theta_1),\theta_2) \\ G(x) = y + f_3(y, \theta_3) \end{cases}$

$$\star\ \frac{\partial F(x)}{\partial \theta_1} = \frac{\partial f_3(y, \theta_3)}{\partial \theta_1} \underset{\text{(chain Rule)}}{=} \frac{\partial f_3(y,\theta_3)}{\partial y} \times \frac{\partial y}{\partial \theta_1}$$

$$= \frac{\partial f_3(y,\theta_3)}{\partial y} \times \frac{\partial f_2(f_1(x,\theta_1),\theta_2)}{\partial \theta_1}$$

$$= \frac{\partial f_3(y,\theta_3)}{\partial y} \times \frac{\partial f_2(z,\theta_2)}{\partial z} \times \frac{\partial f_1(x,\theta_1)}{\partial \theta_1}$$

$$\star\ \frac{\partial F(x)}{\partial \theta_2} = \frac{\partial f_3(y,\theta_3)}{\partial \theta_2} \underset{\text{(chain Rule)}}{=} \frac{\partial f_3(y,\theta_3)}{\partial y} \times \frac{\partial f_2(f_1(x,\theta_1),\theta_2)}{\partial \theta_2}$$

* $$\frac{\partial F(x)}{\partial \theta_2} = \frac{\partial f_3(y, \theta_3)}{\partial y} \times \frac{\partial f_2(z, \theta_2)}{\partial \theta_2} \; ; \; z = f_1(x, \theta_1)$$

* $$\frac{\partial F(x)}{\partial \theta_3} = \frac{\partial f_3(y, \theta_3)}{\partial \theta_3} \; ; \; y = f_2(f_1(x, \theta_1), \theta_2)$$

* $$\frac{\partial G(x)}{\partial \theta_1} = \frac{\partial y + f_3(y, \theta_3)}{\partial \theta_1} \overset{\text{(chain Rule)}}{=} \frac{\partial y + f_3(y, \theta_3)}{\partial y} \times \frac{\partial y}{\partial \theta_1}$$
$$= \left(1 + \frac{\partial f_3(y, \theta_3)}{\partial y}\right)\left(\frac{\partial f_2(z, \theta_2)}{\partial z} \times \frac{\partial f_1(x, \theta_1)}{\partial \theta_1}\right)$$

* $$\frac{\partial G(x)}{\partial \theta_2} = \frac{\partial y + f_3(y, \theta_3)}{\partial \theta_2} \overset{\text{(chain Rule)}}{=} \frac{\partial y + f_3(y, \theta_3)}{\partial y} \times \frac{\partial y}{\partial \theta_2}$$
$$= \left(1 + \frac{\partial f_3(y, \theta_3)}{\partial y}\right) \times \frac{\partial f_2(z, \theta_2)}{\partial \theta_2}$$
with $z = f_1(x, \theta_1)$

* $$\frac{\partial G(x)}{\partial \theta_3} = \frac{\partial y + f_3(y, \theta_3)}{\partial \theta_3} = \frac{\partial f_3(y, \theta_3)}{\partial \theta_3}$$

$\Rightarrow$ By adding a skipping term, we are adding a constant equal to 1 to the gradient of $\frac{\partial f_3}{\partial y}$. This result will help reducing the vanishing gradient when arriving at the last layer of the network.