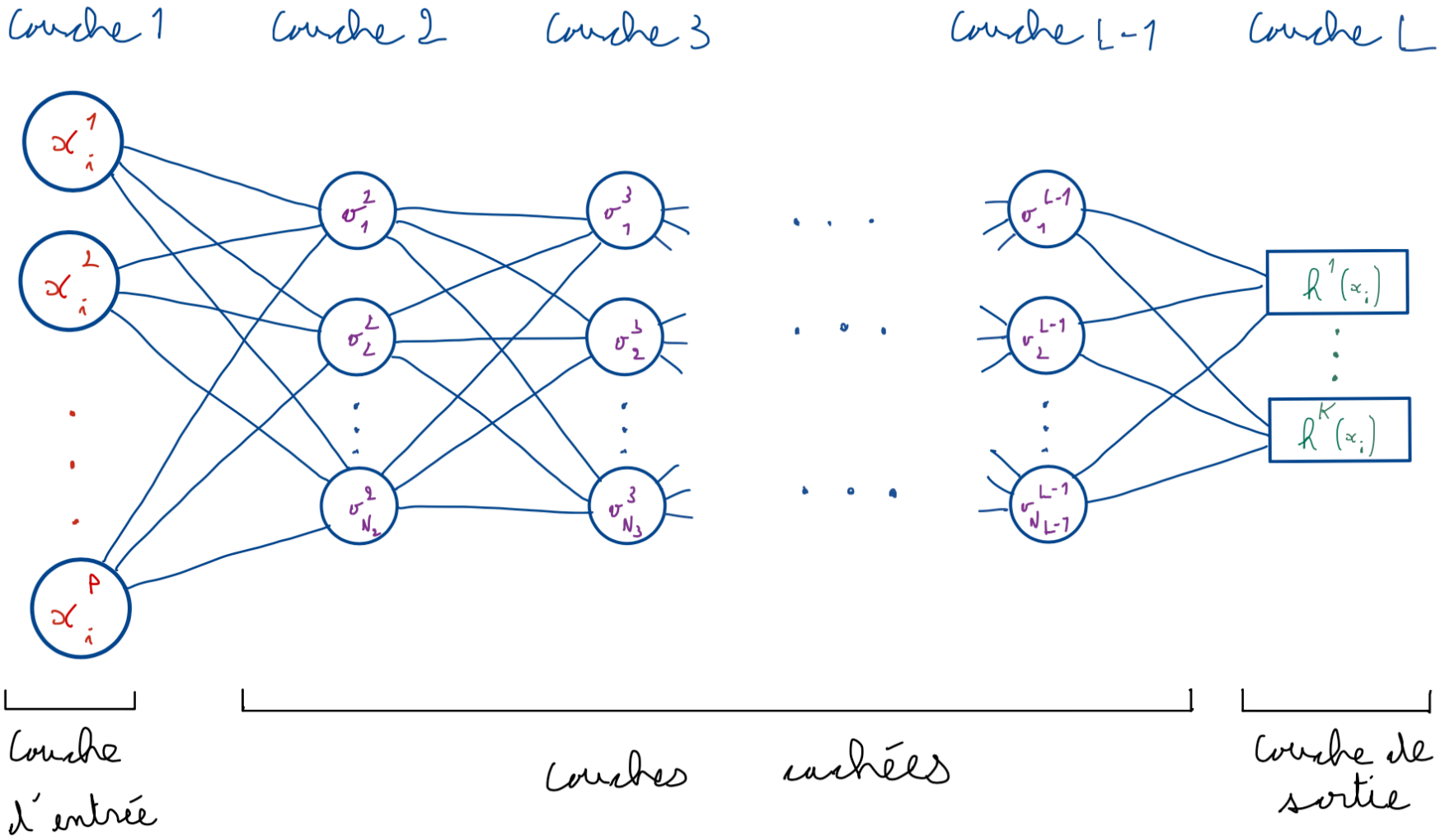
Introduction au calcul GP-GPU

Partie 3 : Application aux réseaux de neurones convolutifs

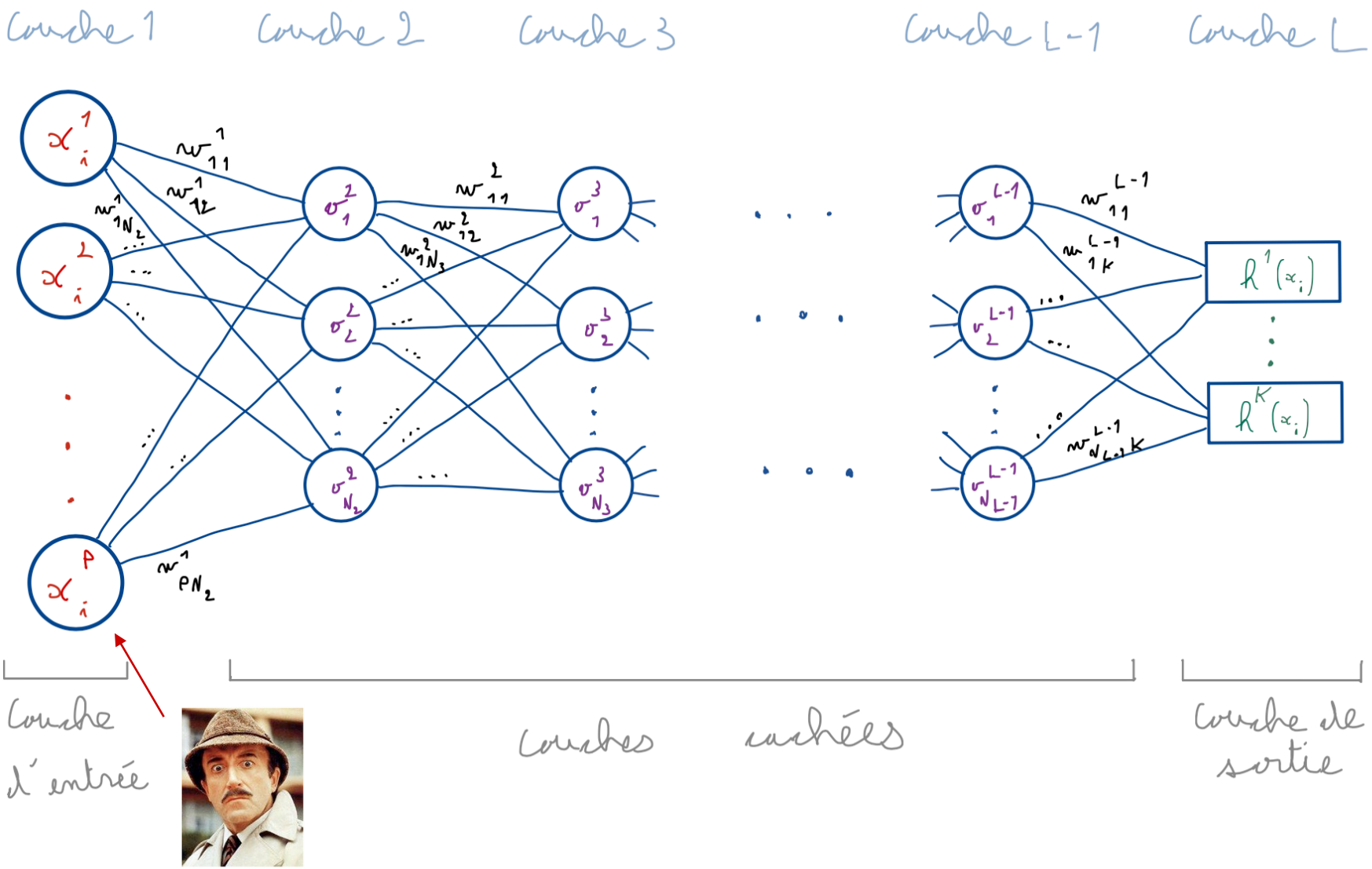
Laurent Risser

Institut de mathématiques de Toulouse
lrisser@math.univ-toulouse.fr

1) Calcul d'un gradient dans un réseau de neurones



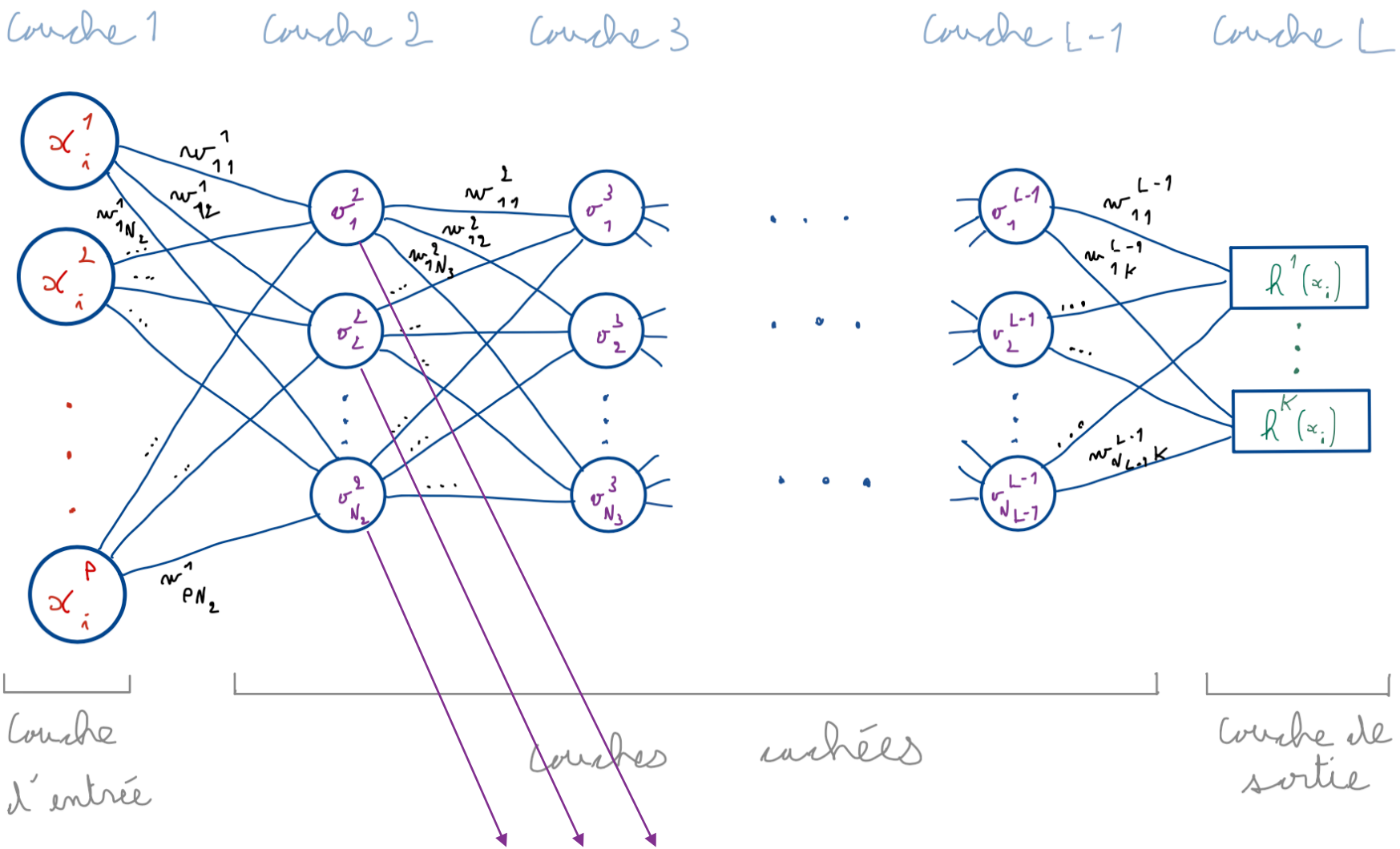
1) Calcul d'un gradient dans un réseau de neurones



$$\forall k = 1, \ldots, p :$$

$$o_k^1 = x_i^k$$

1) Calcul d'un gradient dans un réseau de neurones



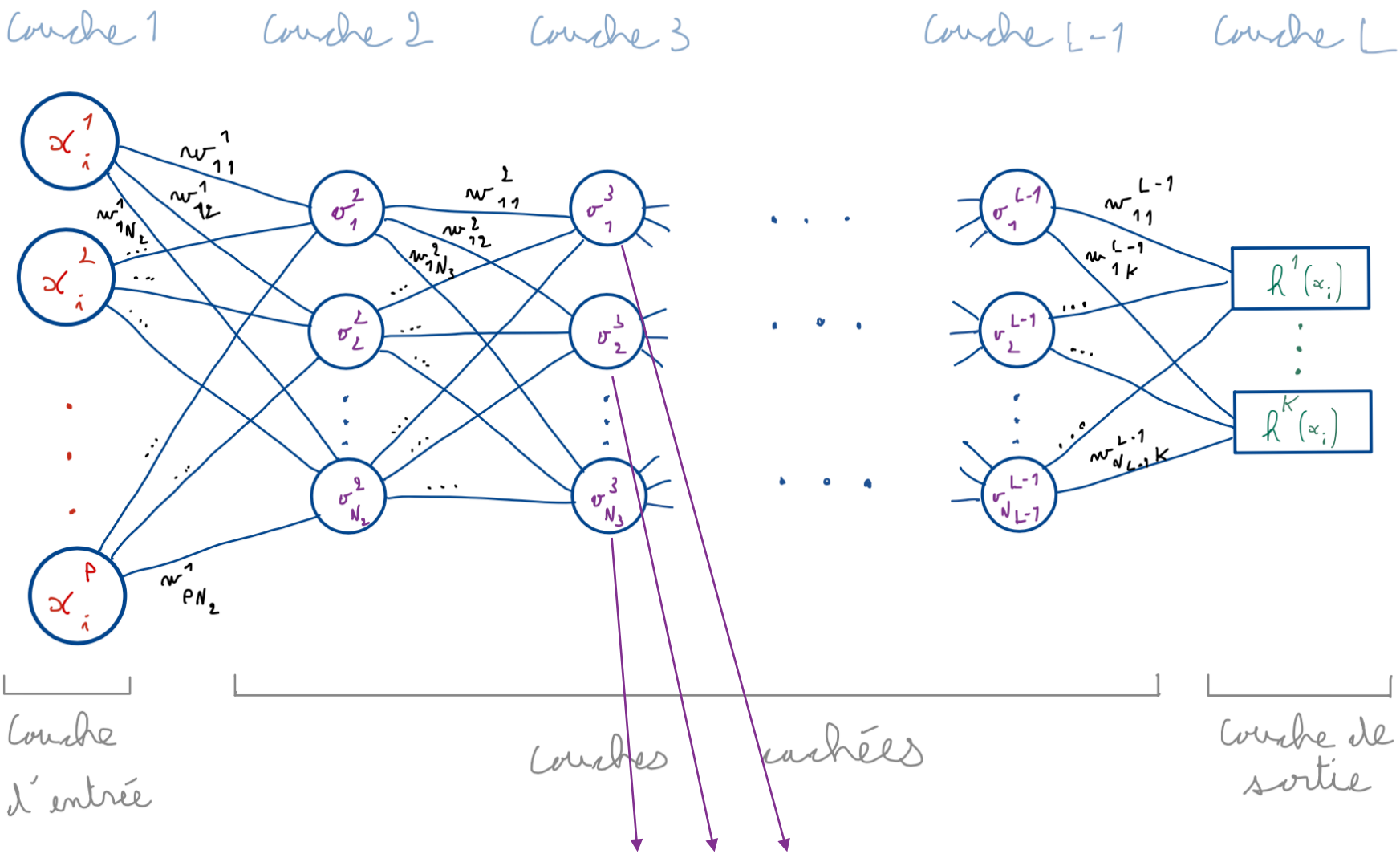
$\forall k = 1,\dots,p$:

$o_k^1 = x_i^k$

$$\begin{cases} s_k^{l+1} = \sum_{p\in\{1,\dots,N_l\}} w_{pk}^l o_p^l \\ o_k^{l+1} = \varphi(s_k^{l+1}) \end{cases}$$

$\varphi(\,.\,)$ non-linéaire $\rightarrow$ ici $\varphi(s_k^{l+1}) = ReLU(s_k^{l+1}) = \max(0\,,s_k^{l+1})$

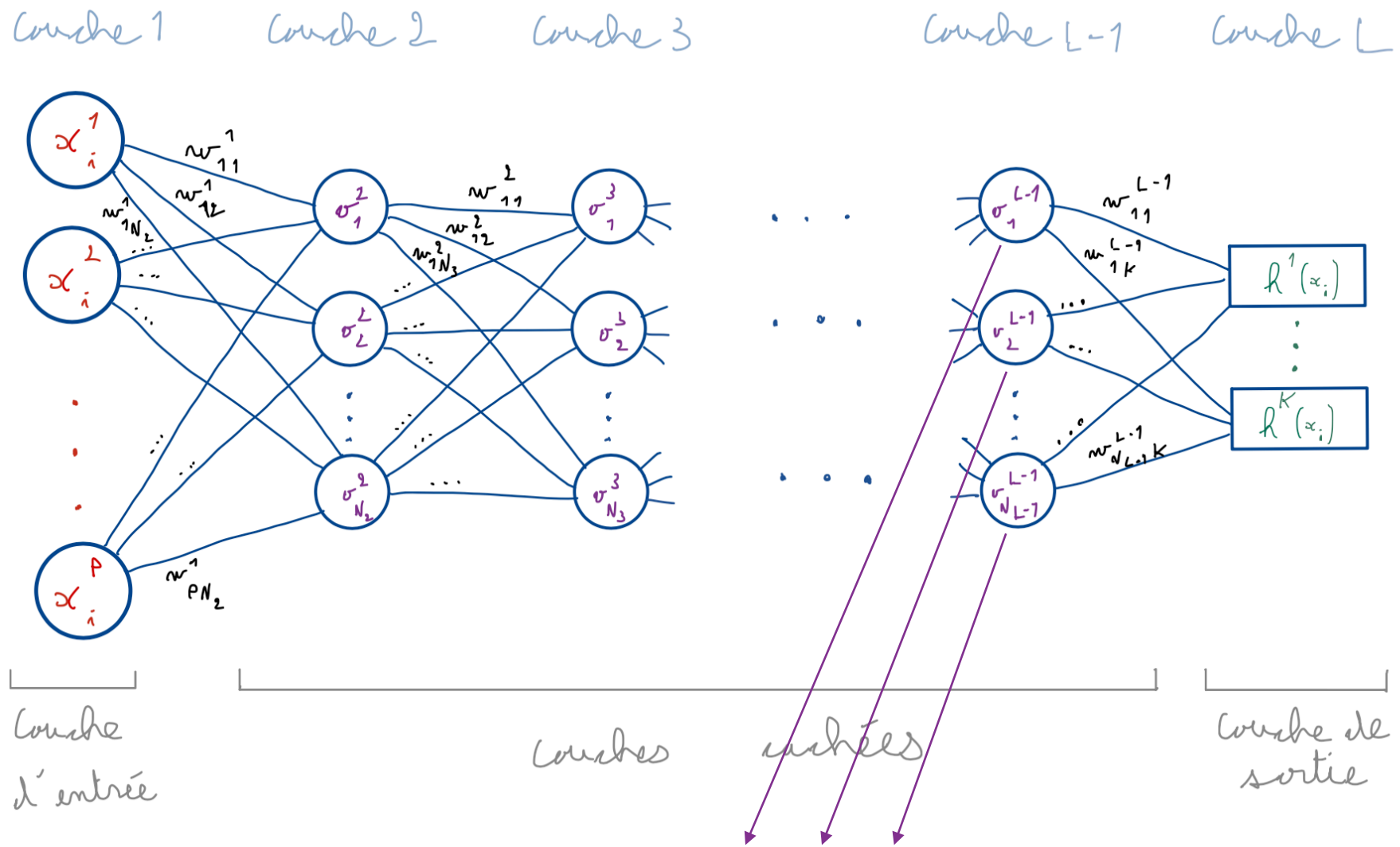
1) Calcul d'un gradient dans un réseau de neurones



$$\begin{cases} s_k^{l+1} = \sum_{p\in\{1,\dots,N_l\}} w_{pk}^l o_p^l \\ o_k^{l+1} = \varphi(s_k^{l+1}) \end{cases}$$

$\varphi(\,.\,)$ non-linéaire $\rightarrow$ ici $\varphi(s_k^{l+1}) = ReLU(s_k^{l+1}) = \max(0\,, s_k^{l+1})$

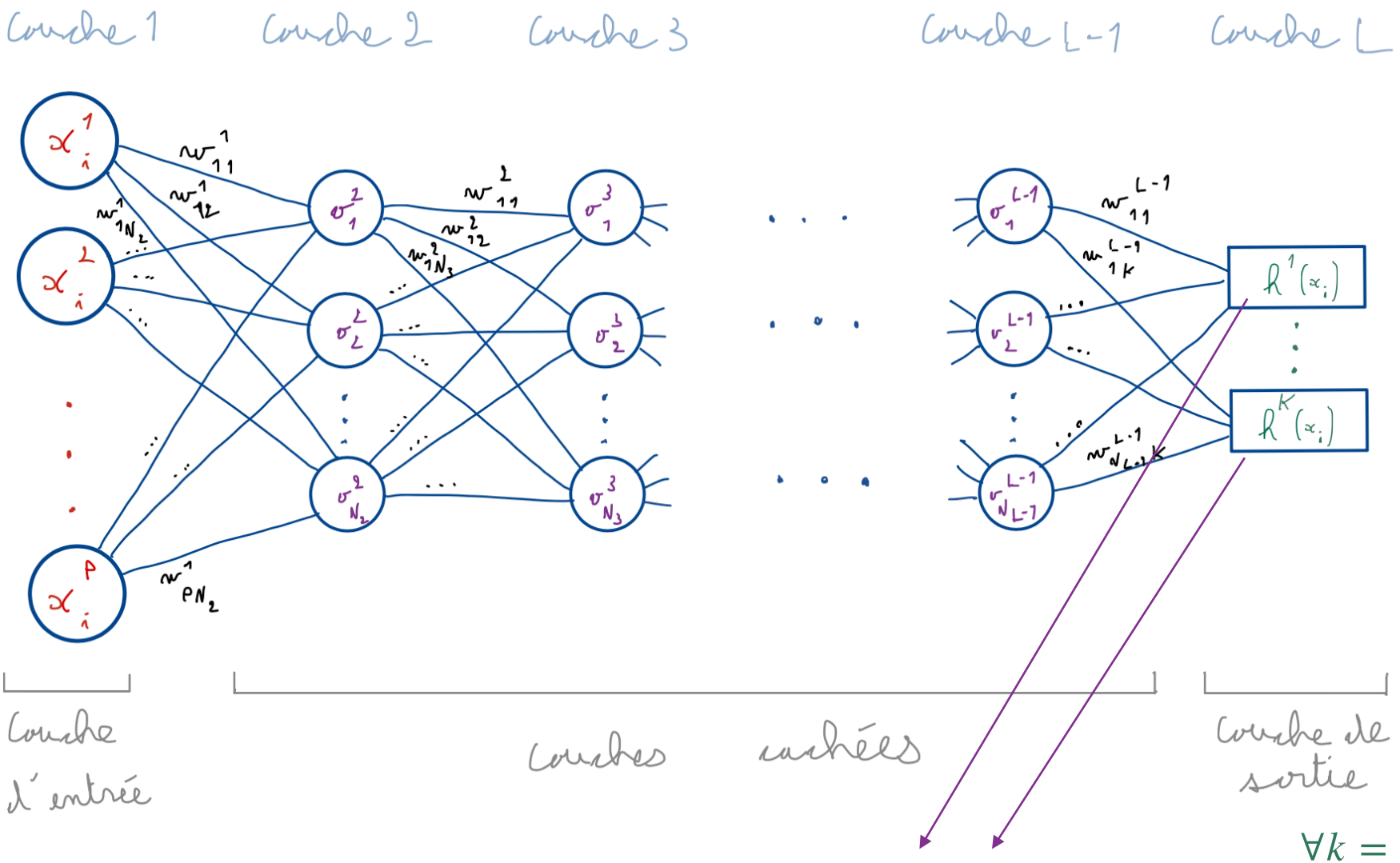
1) Calcul d'un gradient dans un réseau de neurones



$$\begin{cases} s_k^{l+1} = \sum_{p\in\{1,\dots,N_l\}} w_{pk}^l o_p^l \\ o_k^{l+1} = \varphi(s_k^{l+1}) \end{cases}$$

$\varphi(\,.\,)$ non-linéaire $\rightarrow$ ici $\varphi(s_k^{l+1}) = ReLU(s_k^{l+1}) = \max(0\,, s_k^{l+1})$

1) Calcul d'un gradient dans un réseau de neurones

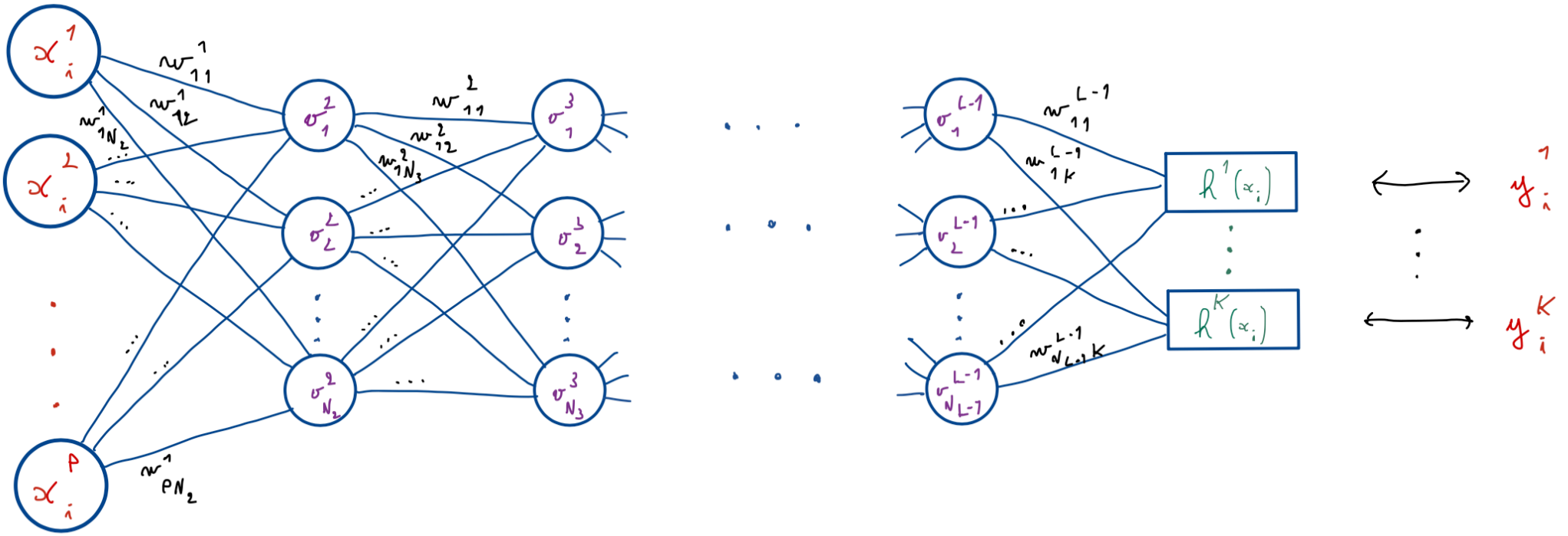


$$\begin{cases} s_k^{l+1} = \sum_{p\in\{1,\dots,N_l\}} w_{pk}^l o_p^l \\ o_k^{l+1} = \varphi(s_k^{l+1}) \end{cases}$$

$\forall k = 1,\dots,K :$
$h^k(x_i) = o_k^L$

$\varphi(\,.\,)$ non-linéaire $\rightarrow$ ici $\varphi(s_k^{l+1}) = logistic(s_k^{l+1}) = 1/\left(1+e^{-s_k^{l+1}}\right)$

1) Calcul d'un gradient dans un réseau de neurones



Couche de sortie

Vrai sorties

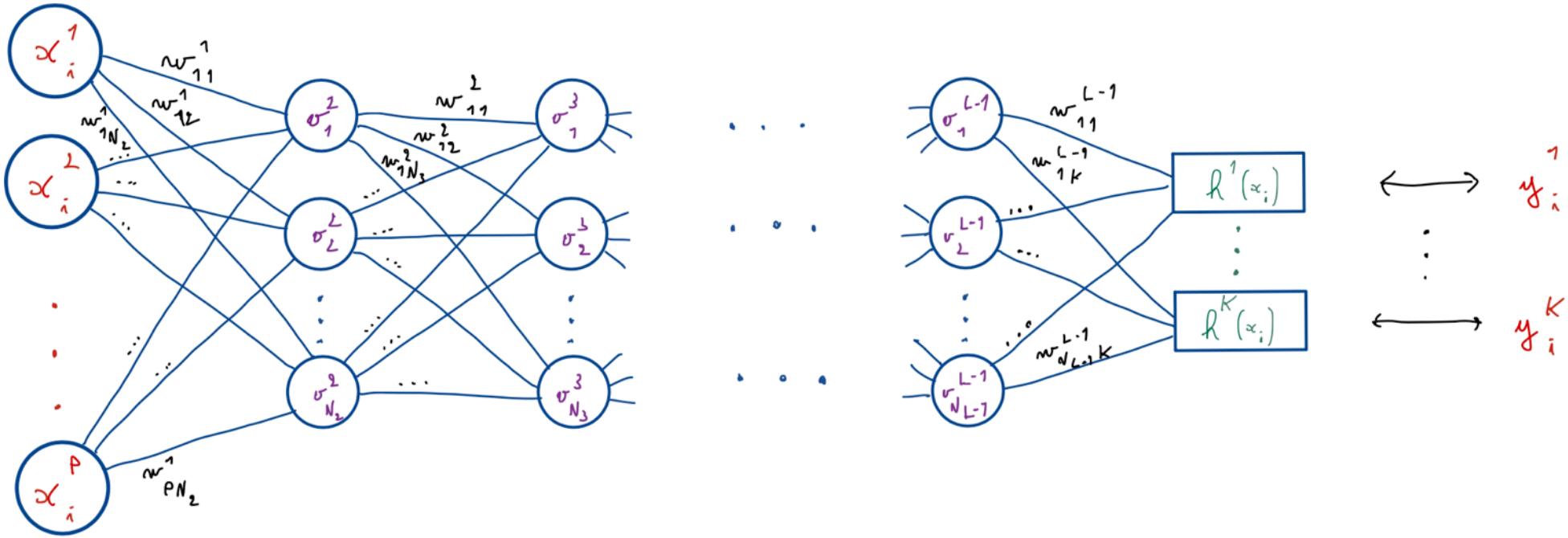
Loss possible : $loss\left(h_\Theta(x_i), y_i\right) = \sum_{k=1}^{K} (h_\Theta^k(x_i) - y_i^k)^2$

Remarque :

- Il existe d'autres losses comme la C*ross-entropy.*
- Le loss doit être dérivable par rapport aux $h_\Theta(x_i)$

Définition d'un loss en vue de l'apprentissage des paramètres $\Theta = \{w_{pk}^l\}_{p,k,l}$

1) Calcul d'un gradient dans un réseau de neurones



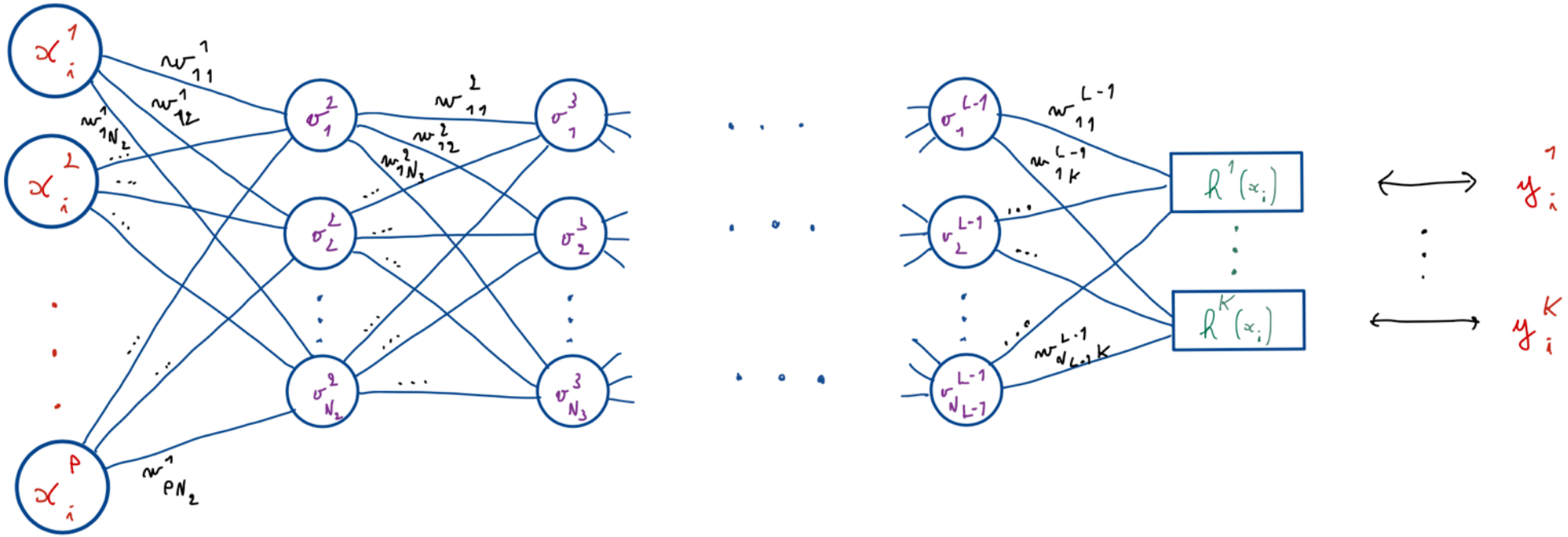
Recherche des paramètres optimaux $\hat{\Theta}$ tels que :

$$\hat{\Theta} = \arg\, min_{\Theta=\{w_0,\ldots,w_p\}} R_\Theta\left((x_i, y_i)_{i=1,\ldots,n}\right)$$

$$= \arg\, min_{\Theta=\{w_0,\ldots,w_p\}} \frac{1}{n}\sum_{i=1}^{n} loss\left(h_\Theta(x_i), y_i\right)$$

→ **descente de gradient en grande dimension**

1) Calcul d'un gradient dans un réseau de neurones



$$\nabla R_\Theta\left((x_i, y_i)_{i=1,\dots,n}\right) = \left(\frac{\partial R_\Theta(\dots)}{\partial w^1_{1,1}}, \frac{\partial R_\Theta(\dots)}{\partial w^1_{1,2}}, \dots, \frac{\partial R_\Theta(\dots)}{\partial w^{N_{L-1}}_{N_{L-1},K}}\right)^\intercal$$

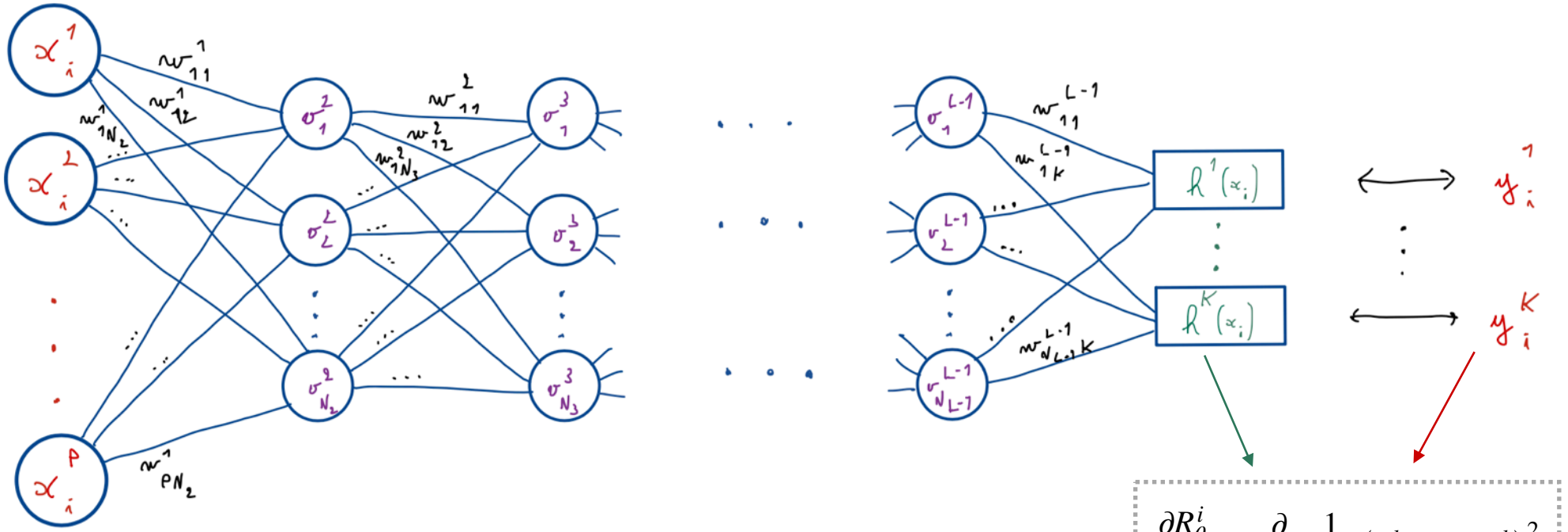
(Gradient du risque empirique)

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\left(\frac{\partial loss(h^k(x_i), y_i^k)}{\partial w^1_{1,1}}, \frac{\partial loss(h^k(x_i), y_i^k)}{\partial w^1_{1,2}}, \dots, \frac{\partial loss(h^k(x_i), y_i^k)}{\partial w^{N_{L-1}}_{N_{L-1},K}}\right)^\intercal$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\left(\frac{\partial\left(h^k(x_i) - y_i^k\right)^2}{\partial w^1_{1,1}}, \frac{\partial\left(h^k(x_i) - y_i^k\right)^2}{\partial w^1_{1,2}}, \dots, \frac{\partial\left(h^k(x_i) - y_i^k\right)^2}{\partial w^{N_{L-1}}_{N_{L-1},K}}\right)^\intercal$$

Calcul analytique possible mais les $h^k(x_i)$ sont issus de fonctions composées couches après couches
→ **dérivation avec le théorème des fonctions composées (*chain rule*) !**

1) Calcul d'un gradient dans un réseau de neurones

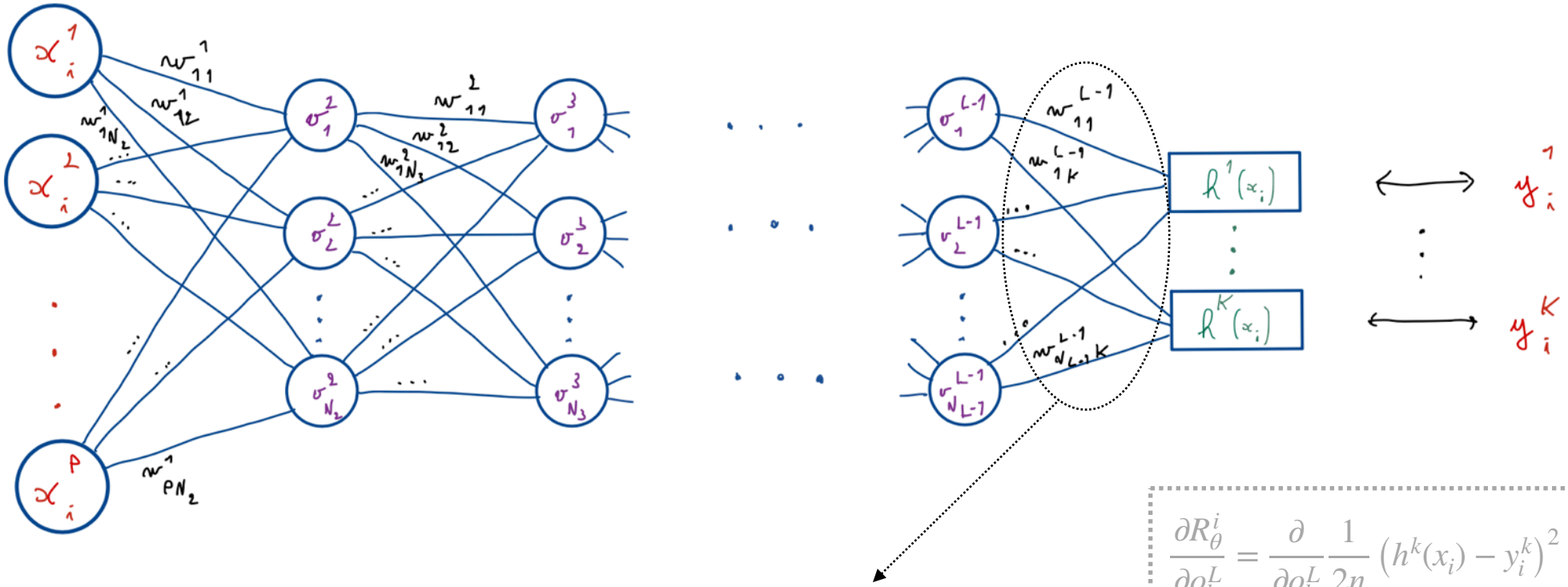


$$\frac{\partial R_\theta^i}{\partial o_k^L} = \frac{\partial}{\partial o_k^L}\frac{1}{2n}\left(h^k(x_i) - y_i^k\right)^2$$
$$= \frac{1}{n}\left(h^k(x_i) - y_i^k\right)$$

Remarques :

- Notation simplifiée $R_\theta^i = loss(h(x_i), y_i)$
- $h^k(x_i)$ est la sortie o_k^L pour l'observation i

1) Calcul d'un gradient dans un réseau de neurones

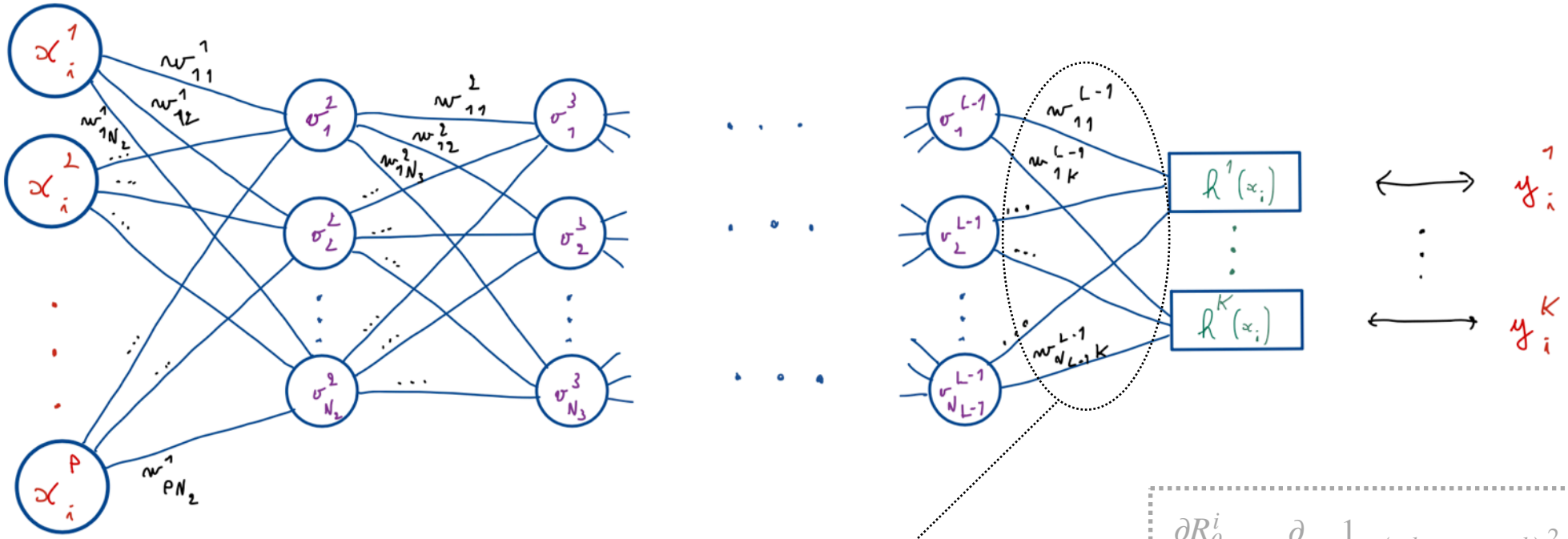


$$\frac{\partial R^i_\Theta}{\partial w^l_{pk}} = \frac{\partial R^i_\Theta}{\partial o^{l+1}_k} \frac{\partial o^{l+1}_k}{\partial s^{l+1}_k} \frac{\partial s^{l+1}_k}{\partial w^l_{pk}}$$

$l = L - 1$

$$\frac{\partial R^i_\theta}{\partial o^L_k} = \frac{\partial}{\partial o^L_k} \frac{1}{2n} \left(h^k(x_i) - y^k_i\right)^2 = \frac{1}{n}\left(h^k(x_i) - y^k_i\right)$$

1) Calcul d'un gradient dans un réseau de neurones



$$\frac{\partial R^i_\Theta}{\partial w^l_{pk}} = \frac{\partial R^i_\Theta}{\partial o^{l+1}_k} \frac{\partial o^{l+1}_k}{\partial s^{l+1}_k} \frac{\partial s^{l+1}_k}{\partial w^l_{pk}}$$

$l = L-1$

$$\frac{\partial R^i_\theta}{\partial o^L_k} = \frac{\partial}{\partial o^L_k} \frac{1}{2n} \left(h^k(x_i) - y^k_i\right)^2 = \frac{1}{n}\left(h^k(x_i) - y^k_i\right)$$

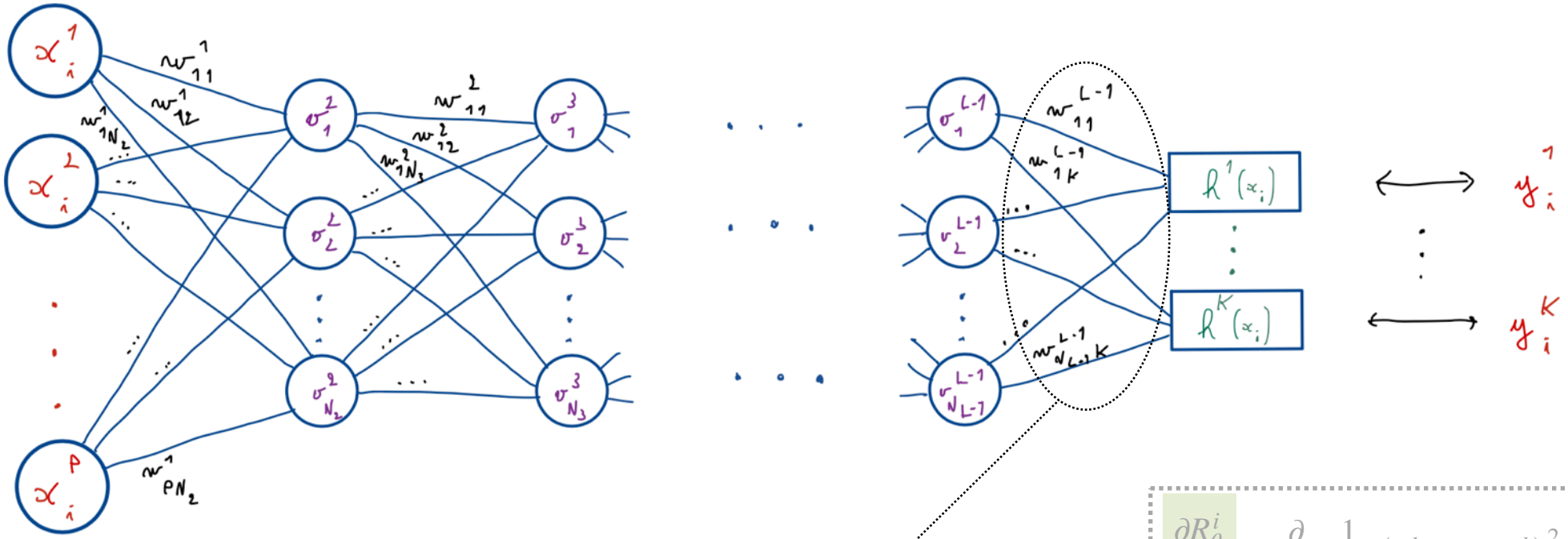
$$\frac{\partial o^{l+1}_k}{\partial s^{l+1}_k} = \frac{\partial}{\partial s^{l+1}_k} \varphi(s^{l+1}_k) \text{ [connu]}$$

$l = L-1$

$$\frac{\partial s^{l+1}_k}{\partial w^l_{pk}} = \frac{\partial}{\partial w^l_{pk}} \sum_{p' \in \{1,\dots,N_l\}} w^l_{p'k} o^l_{p'} = \frac{\partial}{\partial w^l_{pk}} w^l_{pk} o^l_p = o^l_p$$

$l = L-1$

1) Calcul d'un gradient dans un réseau de neurones



$$\frac{\partial R^i_\Theta}{\partial w^l_{pk}} = \frac{\partial R^i_\Theta}{\partial o^{l+1}_k} \frac{\partial o^{l+1}_k}{\partial s^{l+1}_k} \frac{\partial s^{l+1}_k}{\partial w^l_{pk}}$$

$l = L-1$

$$\frac{\partial R^i_\theta}{\partial o^L_k} = \frac{\partial}{\partial o^L_k} \frac{1}{2n} \left(h^k(x_i) - y^k_i\right)^2 = \frac{1}{n}\left(h^k(x_i) - y^k_i\right)$$

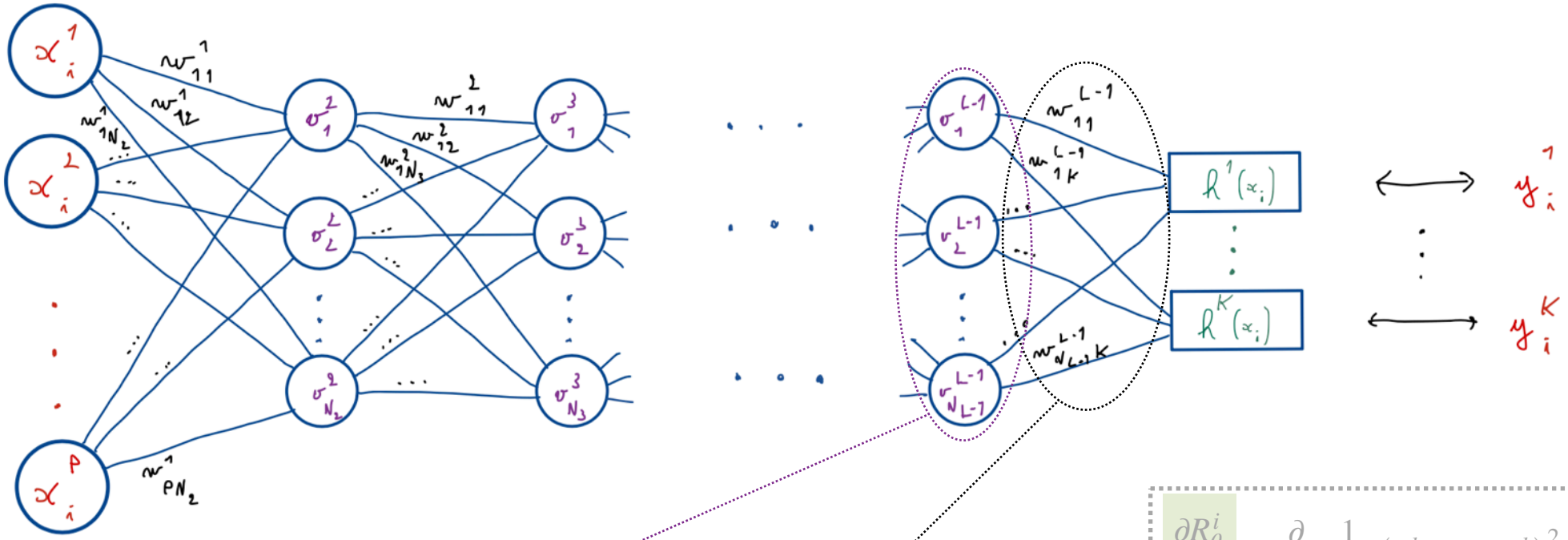
$$\frac{\partial o^{l+1}_k}{\partial s^{l+1}_k} = \frac{\partial}{\partial s^{l+1}_k} \varphi(s^{l+1}_k) \text{ [connu]}$$

$l = L-1$

$$\frac{\partial s^{l+1}_k}{\partial w^l_{pk}} = \frac{\partial}{\partial w^l_{pk}} \sum_{p' \in \{1,\dots,N_l\}} w^l_{p'k} o^l_{p'} = \frac{\partial}{\partial w^l_{pk}} w^l_{pk} o^l_p = o^l_p$$

$l = L-1$

1) Calcul d'un gradient dans un réseau de neurones



$$\frac{\partial R^i_\Theta}{\partial o^l_k} = \sum_{p=1}^{N_{l+1}} \underbrace{\frac{\partial R^i_\Theta}{\partial o^{l+1}_k}}_{\text{Connu à la couche } l+1} \underbrace{\frac{\partial o^{l+1}_k}{\partial s^{l+1}_k}}_{\text{Derivee de } \varphi} \underbrace{\frac{\partial s^{l+1}_k}{\partial o^l_k}}_{=w^l_{pk}}$$

$l = L-1$

$$\frac{\partial R^i_\Theta}{\partial w^l_{pk}} = \frac{\partial R^i_\Theta}{\partial o^{l+1}_k} \frac{\partial o^{l+1}_k}{\partial s^{l+1}_k} \frac{\partial s^{l+1}_k}{\partial w^l_{pk}}$$

$l = L-1$

$$\frac{\partial R^i_\theta}{\partial o^L_k} = \frac{\partial}{\partial o^L_k} \frac{1}{2n} \left(h^k(x_i) - y^k_i\right)^2 = \frac{1}{n}\left(h^k(x_i) - y^k_i\right)$$

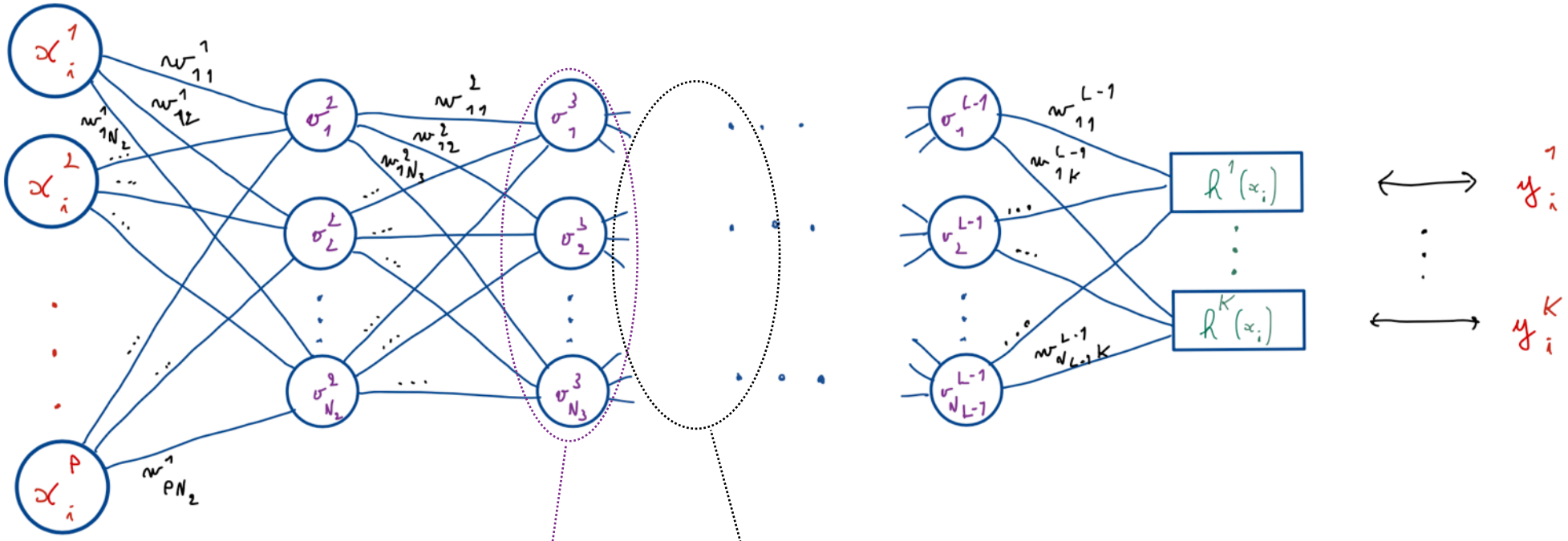
$$\frac{\partial o^{l+1}_k}{\partial s^{l+1}_k} = \frac{\partial}{\partial s^{l+1}_k} \varphi(s^{l+1}_k) \text{ [connu]}$$

$l = L-1$

$$\frac{\partial s^{l+1}_k}{\partial w^l_{pk}} = \frac{\partial}{\partial w^l_{pk}} \sum_{p' \in \{1,\dots,N_l\}} w^l_{p'k} o^l_{p'} = \frac{\partial}{\partial w^l_{pk}} w^l_{pk} o^l_p = o^l_p$$

$l = L-1$

1) Calcul d'un gradient dans un réseau de neurones



$$\frac{\partial R_\Theta^i}{\partial o_k^l} = \sum_{p=1}^{N_{l+1}} \underbrace{\frac{\partial R_\Theta^i}{\partial o_k^{l+1}}}_{\text{Connu à la couche } l+1} \underbrace{\frac{\partial o_k^{l+1}}{\partial s_k^{l+1}}}_{\text{Derivee de } \varphi} \underbrace{\frac{\partial s_k^{l+1}}{\partial o_k^l}}_{=w_{pk}^l}$$

$l = 3$

$$\frac{\partial R_\Theta^i}{\partial w_{pk}^l} = \frac{\partial R_\Theta^i}{\partial o_k^{l+1}} \frac{\partial o_k^{l+1}}{\partial s_k^{l+1}} \frac{\partial s_k^{l+1}}{\partial w_{pk}^l}$$

$l = 3$

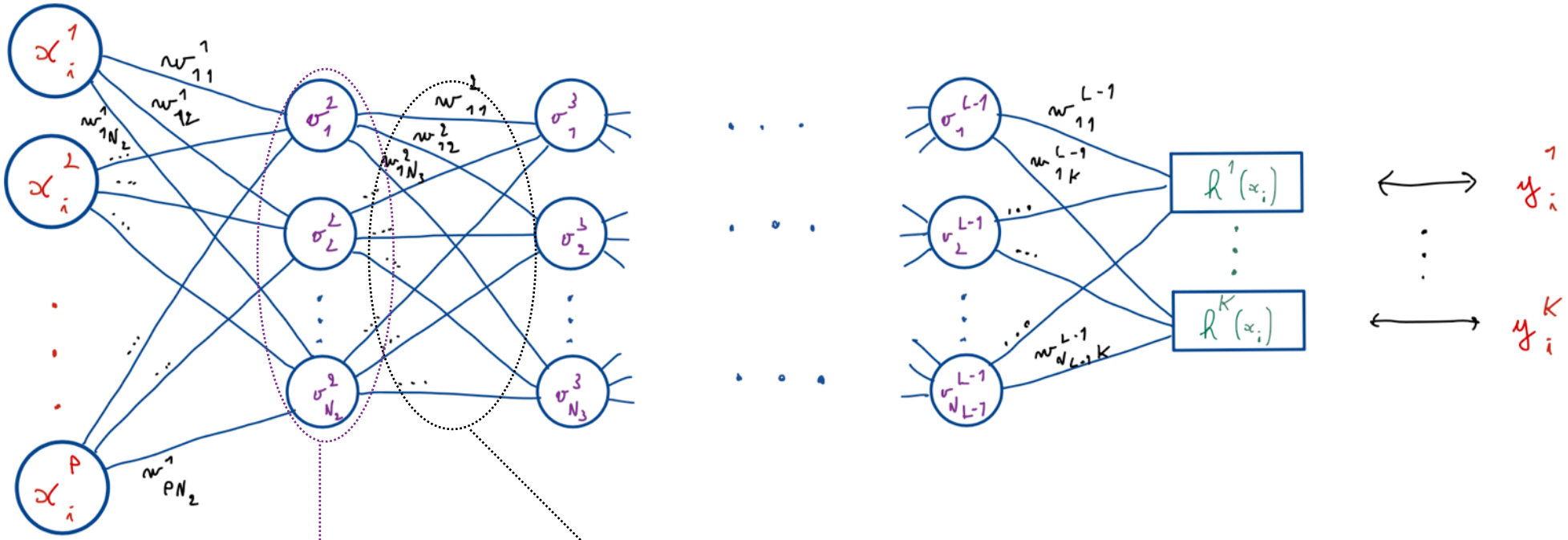
$$\frac{\partial o_k^{l+1}}{\partial s_k^{l+1}} = \frac{\partial}{\partial s_k^{l+1}} \varphi(s_k^{l+1}) \text{ [connu]}$$

$l = 3$

$$\begin{aligned} \frac{\partial s_k^{l+1}}{\partial w_{pk}^l} &= \frac{\partial}{\partial w_{pk}^l} \sum_{p' \in \{1,\dots,N_l\}} w_{p'k}^l o_{p'}^l \\ &= \frac{\partial}{\partial w_{pk}^l} w_{pk}^l o_p^l \\ &= o_p^l \end{aligned}$$

$l = 3$

1) Calcul d'un gradient dans un réseau de neurones



$$\frac{\partial R^i_\Theta}{\partial o^l_k} = \sum_{p=1}^{N_{l+1}} \underbrace{\frac{\partial R^i_\Theta}{\partial o^{l+1}_k}}_{\text{Connu à la couche } l+1} \underbrace{\frac{\partial o^{l+1}_k}{\partial s^{l+1}_k}}_{\text{Derivee de } \varphi} \underbrace{\frac{\partial s^{l+1}_k}{\partial o^l_k}}_{=w^l_{pk}}$$

$l = 2$

$$\frac{\partial R^i_\Theta}{\partial w^l_{pk}} = \frac{\partial R^i_\Theta}{\partial o^{l+1}_k} \frac{\partial o^{l+1}_k}{\partial s^{l+1}_k} \frac{\partial s^{l+1}_k}{\partial w^l_{pk}}$$

$l = 2$

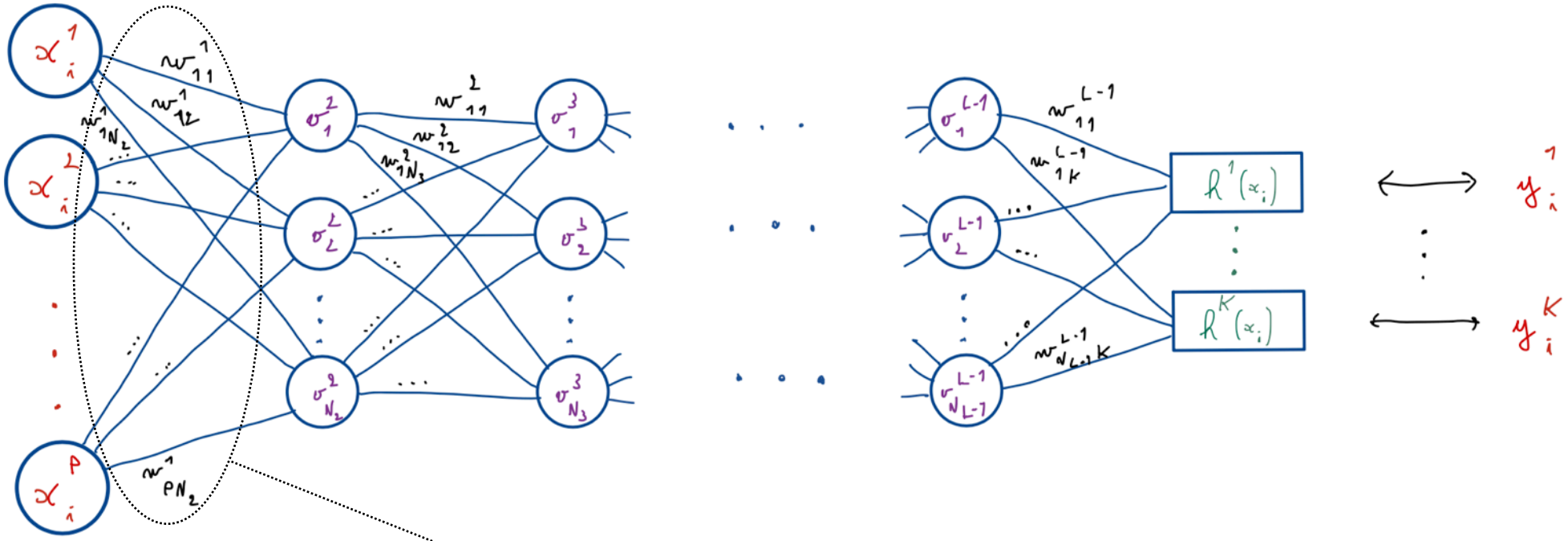
$$\frac{\partial o^{l+1}_k}{\partial s^{l+1}_k} = \frac{\partial}{\partial s^{l+1}_k} \varphi(s^{l+1}_k) \text{ [connu]}$$

$l = 2$

$$\begin{aligned} \frac{\partial s^{l+1}_k}{\partial w^l_{pk}} &= \frac{\partial}{\partial w^l_{pk}} \sum_{p' \in \{1,\dots,N_l\}} w^l_{p'k} o^l_{p'} \\ &= \frac{\partial}{\partial w^l_{pk}} w^l_{pk} o^l_p \\ &= o^l_p \end{aligned}$$

$l = 2$

1) Calcul d'un gradient dans un réseau de neurones



$$\frac{\partial R^i_\Theta}{\partial w^l_{pk}} = \frac{\partial R^i_\Theta}{\partial o^{l+1}_k} \frac{\partial o^{l+1}_k}{\partial s^{l+1}_k} \frac{\partial s^{l+1}_k}{\partial w^l_{pk}}$$

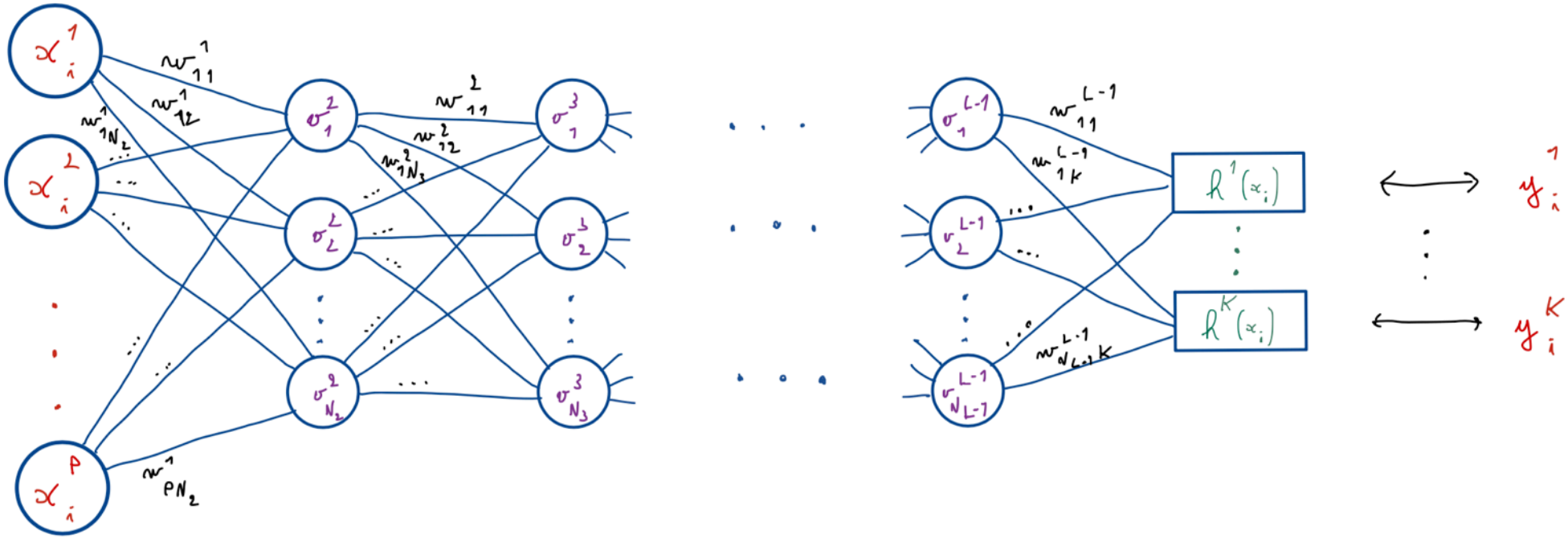
$l = 1$

$$\frac{\partial o^{l+1}_k}{\partial s^{l+1}_k} = \frac{\partial}{\partial s^{l+1}_k} \varphi(s^{l+1}_k) \text{ [connu]}$$

$l = 1$

$$\begin{aligned} \frac{\partial s^{l+1}_k}{\partial w^l_{pk}} &= \frac{\partial}{\partial w^l_{pk}} \sum_{p' \in \{1,\dots,N_l\}} w^l_{p'k} o^l_{p'} \\ &= \frac{\partial}{\partial w^l_{pk}} w^l_{pk} o^l_p \\ &= o^l_p \end{aligned}$$

$l = 1$

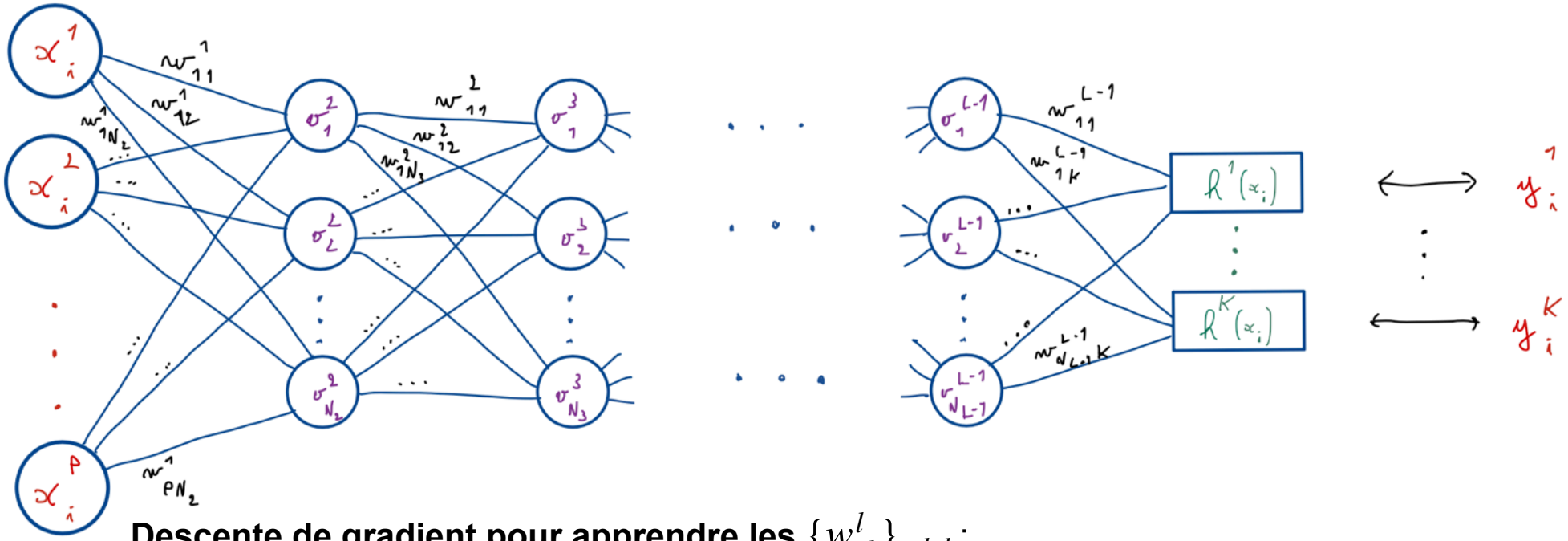


BINGO !!!

On a calculé le gradient pour une observation i :

$$\nabla R_\Theta^i\left((x_i, y_i)_{i=1,\dots,n}\right) = \left(\frac{\partial R_\Theta^i(\dots)}{\partial w_{1,1}^1}, \frac{\partial R_\Theta^i(\dots)}{\partial w_{1,2}^1}, \dots, \frac{\partial R_\Theta^i(\dots)}{\partial w_{N_{L-1},K}^{N_{L-1}}}\right)^\intercal$$

1) Calcul d'un gradient dans un réseau de neurones



Descente de gradient pour apprendre les $\{w_{pk}^l\}_{p,k,l}$:

- (Forward)
  - Pour chaque x_i , calcule $h(x_i) = (h^1(x_i), h^2(x_i), \ldots, h^K(x_i))^\intercal$
  - Pour chaque x_i , calcule $\dfrac{\partial R_\theta^i}{\partial o_k^L}$ sur la couche de sortie en utilisant $h^k(x_i)$ et y_i^k
- (Backward)
  - Pour chaque x_i et pour l de $L-1$ à 1 : Calcule les $\dfrac{\partial R_\theta^i}{\partial w_{pk}^l}$ et $\dfrac{\partial R_\theta^i}{\partial o_p^l}$ en utilisant les $\dfrac{\partial R_\theta^i}{\partial o_p^{l+1}}$
- (Etape de descente de gradient)
  - $\forall \{p,k,l\} : \left(w_{pk}^l\right) = \left(w_{pk}^l\right) - \lambda \dfrac{\partial R_\theta}{\partial w_{pk}^l} = \left(w_{pk}^l\right) - \lambda \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \dfrac{\partial R_\theta^i}{\partial w_{pk}^l}$

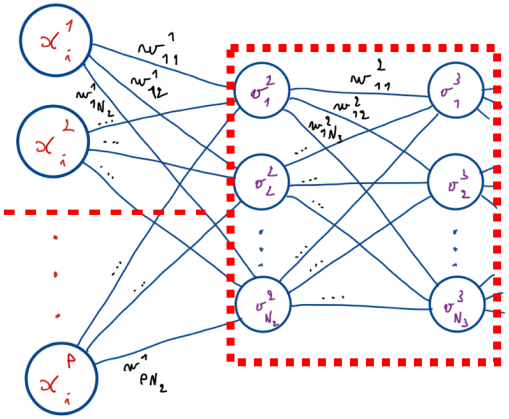
Remarque : Utilise des informations de la phase forward

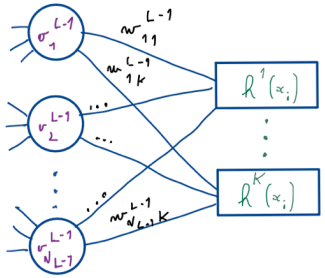
Passage *forward* d'une couche dense d'un réseau de neurones :

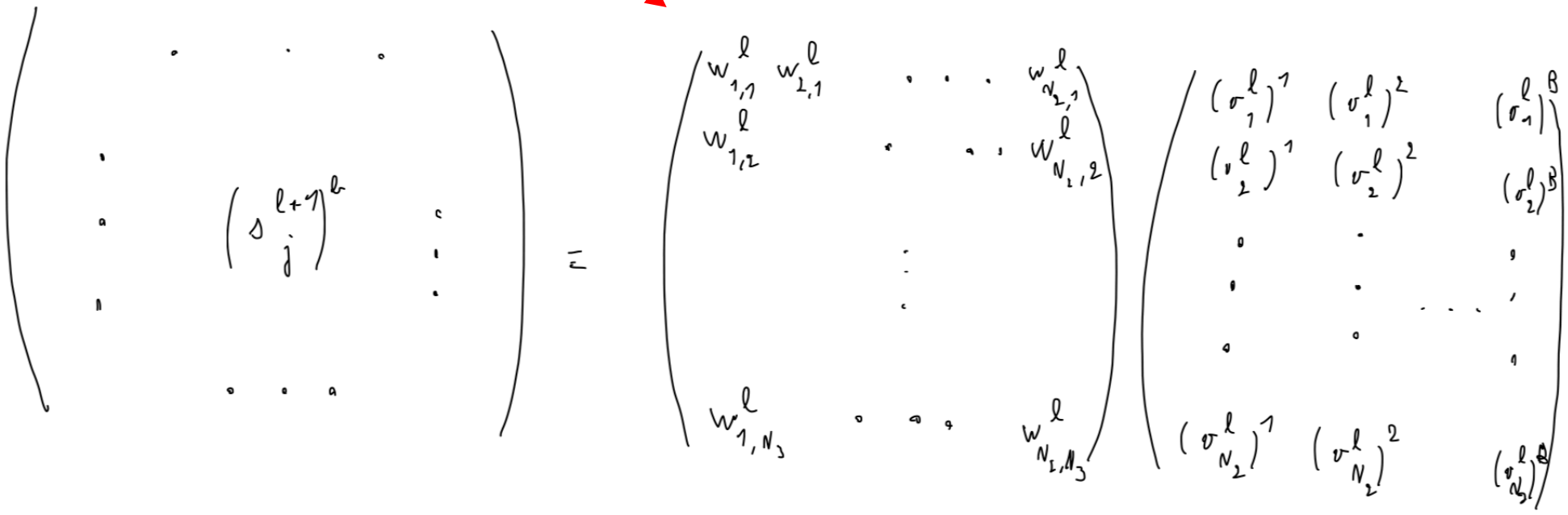
$$\left(s_j^{l+1}\right)^b = \sum_{i=1}^{N_l} \left(o_i^l\right)^b w_{i,j}^l$$

$$\forall j \in \{1,\dots,N_{l+1}\}$$

$$\forall b \in \{1,\dots,B\}$$







Passage *backward* d'une couche dense d'un réseau de neurones :

Tout juste calculé par l'algorithme de backpropagation

Doit être stocké lors de la phase forward

Paramètre du réseau

$$\frac{\partial R_\Theta^i}{\partial \left(o_k^l\right)^b} = \sum_{p=1}^{N_{l+1}} \frac{\partial R_\Theta^i}{\partial \left(o_k^{l+1}\right)^b} \frac{\partial \left(o_k^{l+1}\right)^b}{\partial \left(s_k^{l+1}\right)^b} \frac{\partial \left(s_k^{l+1}\right)^b}{\partial \left(o_k^l\right)^b} = \sum_{p=1}^{N_{l+1}} \frac{\partial R_\Theta^i}{\partial \left(o_k^{l+1}\right)^b} \left(1_{\left(s_k^{l+1}\right)^b \geq 0} \left(w_{pk}\right)^b\right)$$

$$\frac{\partial R_\Theta^i}{\partial \left(w_{pk}^l\right)^b} = \sum_{p=1}^{N_{l+1}} \frac{\partial R_\Theta^i}{\partial \left(o_k^{l+1}\right)^b} \frac{\partial \left(o_k^{l+1}\right)^b}{\partial \left(s_k^{l+1}\right)^b} \frac{\partial \left(s_k^{l+1}\right)^b}{\partial \left(w_{pk}^l\right)^b} = \sum_{p=1}^{N_{l+1}} \frac{\partial R_\Theta^i}{\partial \left(o_k^{l+1}\right)^b} \left(1_{\left(s_k^{l+1}\right)^b \geq 0} \left(o_p^l\right)^b\right)$$

Tout juste calculé par l'algorithme de backpropagation

Doit être stocké lors de la phase forward

Doit être stocké lors de la phase forward

Remarques :
- Multiplications matrices / matrices
- Nécessite des informations stockées lors de la phase *forward*

2) Réseaux convolutifs

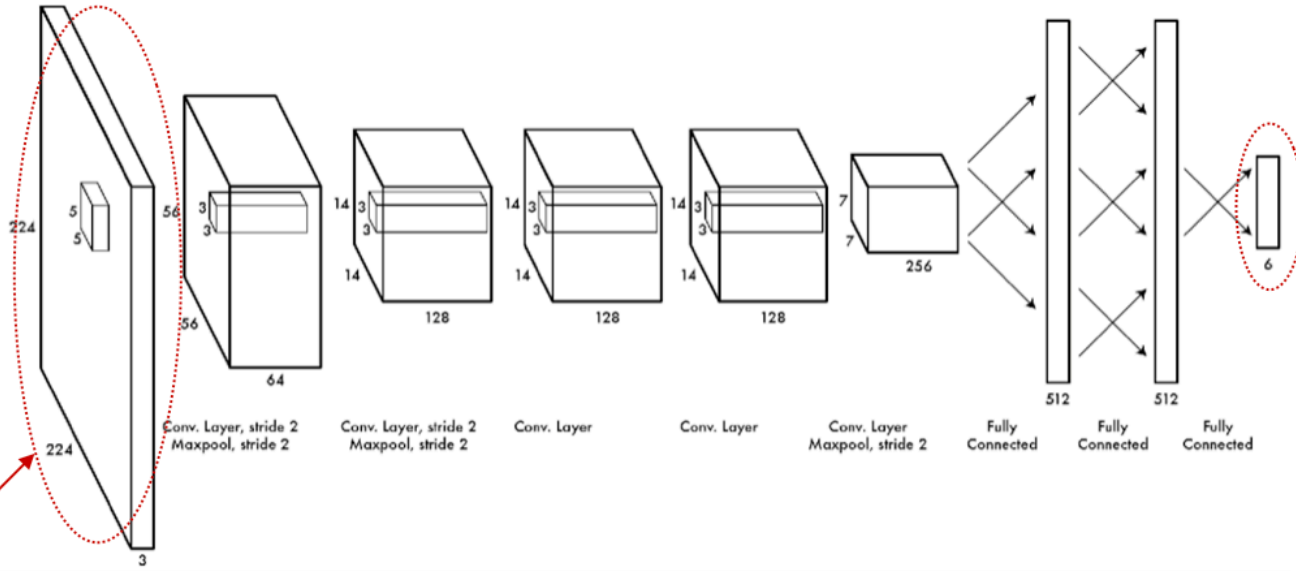


Image RGB de taille 224x224. Exemple x_i =

Couches de convolutions

Couches denses

Probabilité d'avoir 6 caractéristiques dans x_i

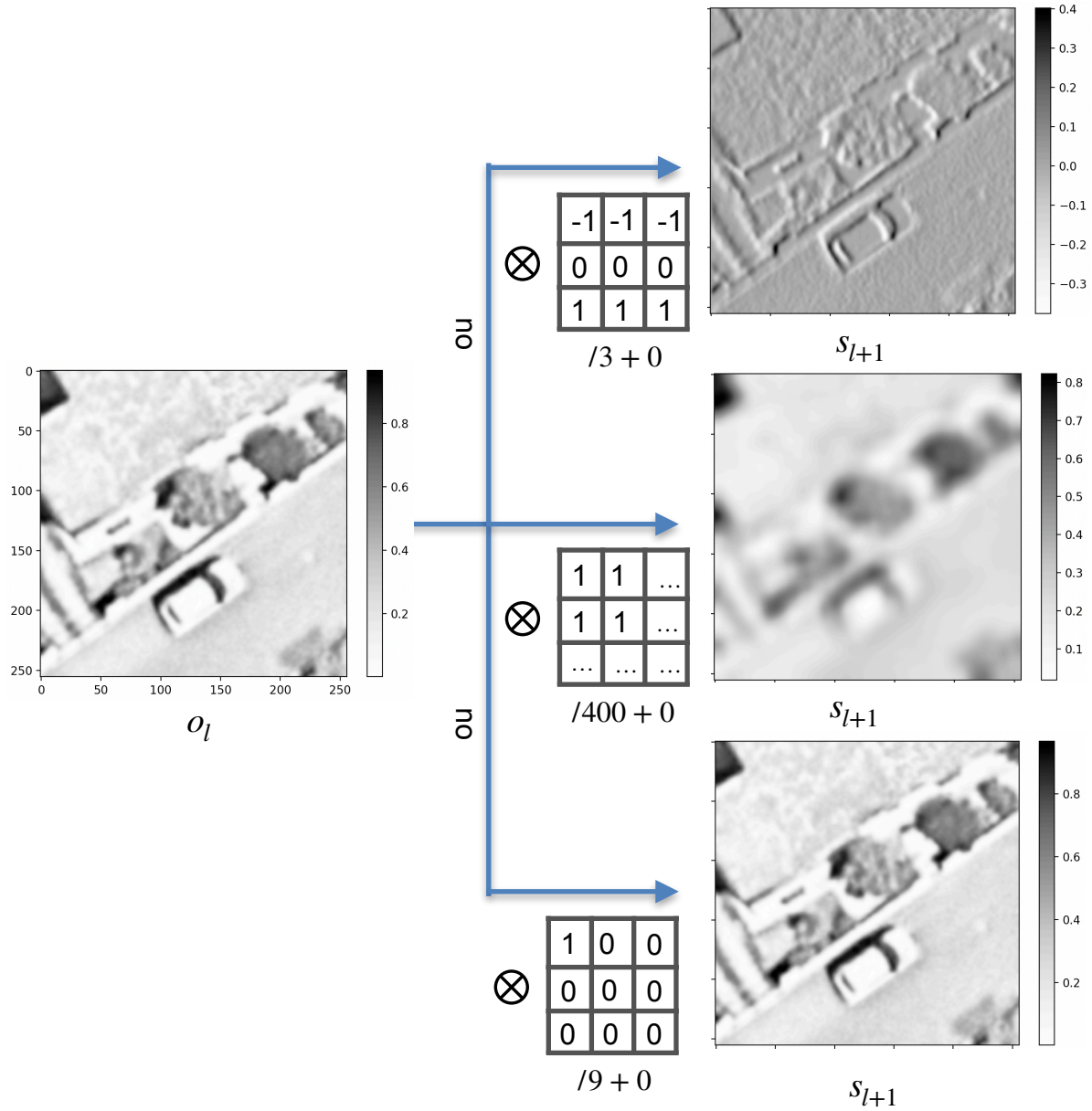
Exemple :

$h_\Theta(x_i) = (0.02,\ldots,0.89)^\intercal$

Lunettes ⋯ Chapeau

Image du réseau : [Redmon and Angelova, IEEE ICRA 2015]

Briques de base : Convolution



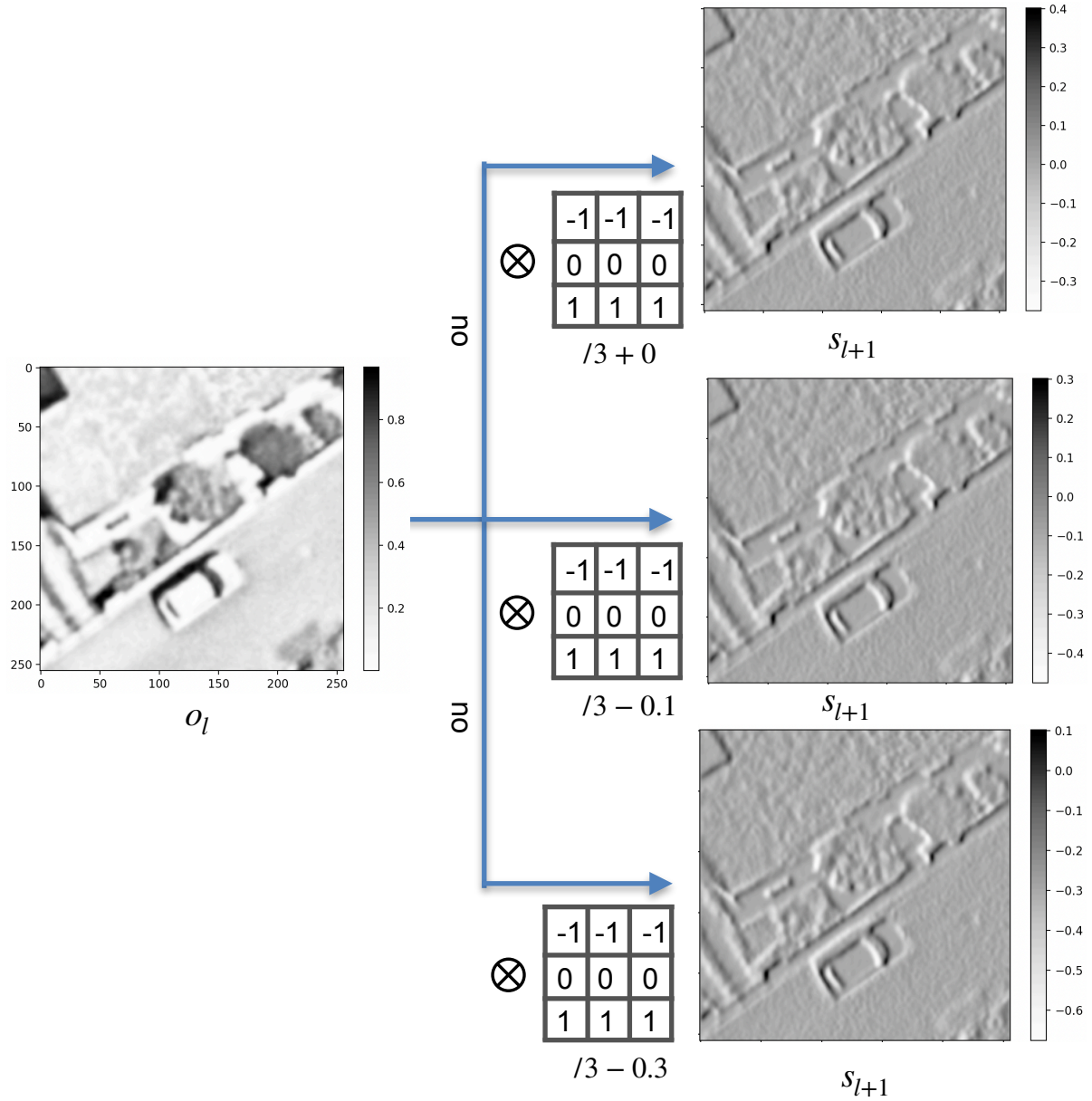
Pixels de l'image — Image filtrée

$$\forall (i,j) \in \Omega : \quad s_{l+1}[i,j] = b + \sum_{(m,n)\in\omega} o_l[i+m, j+n] K[m,n]$$

Biais — Image initiale — Filtre de convolution

Convolution + biais

Briques de base : Convolution et biais



Pixels de l'image

Image filtrée

$$\forall (i,j) \in \Omega : \quad s_{l+1}[i,j] = b + \sum_{(m,n)\in\omega} o_l[i+m, j+n] K[m,n]$$

Biais

Image initiale

Filtre de convolution

Convolution + biais

Briques de base : Convolution et biais suivi de ReLU

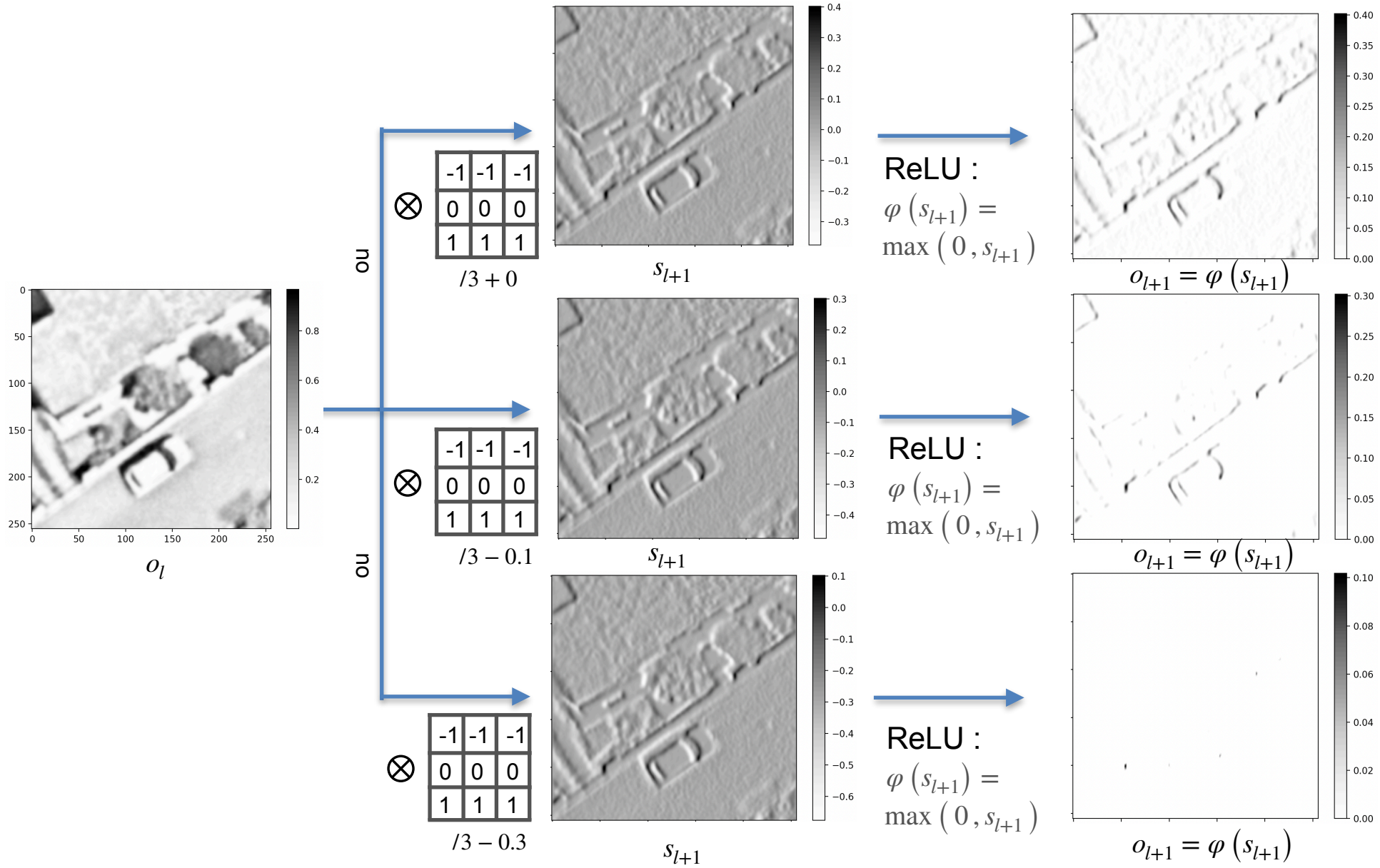
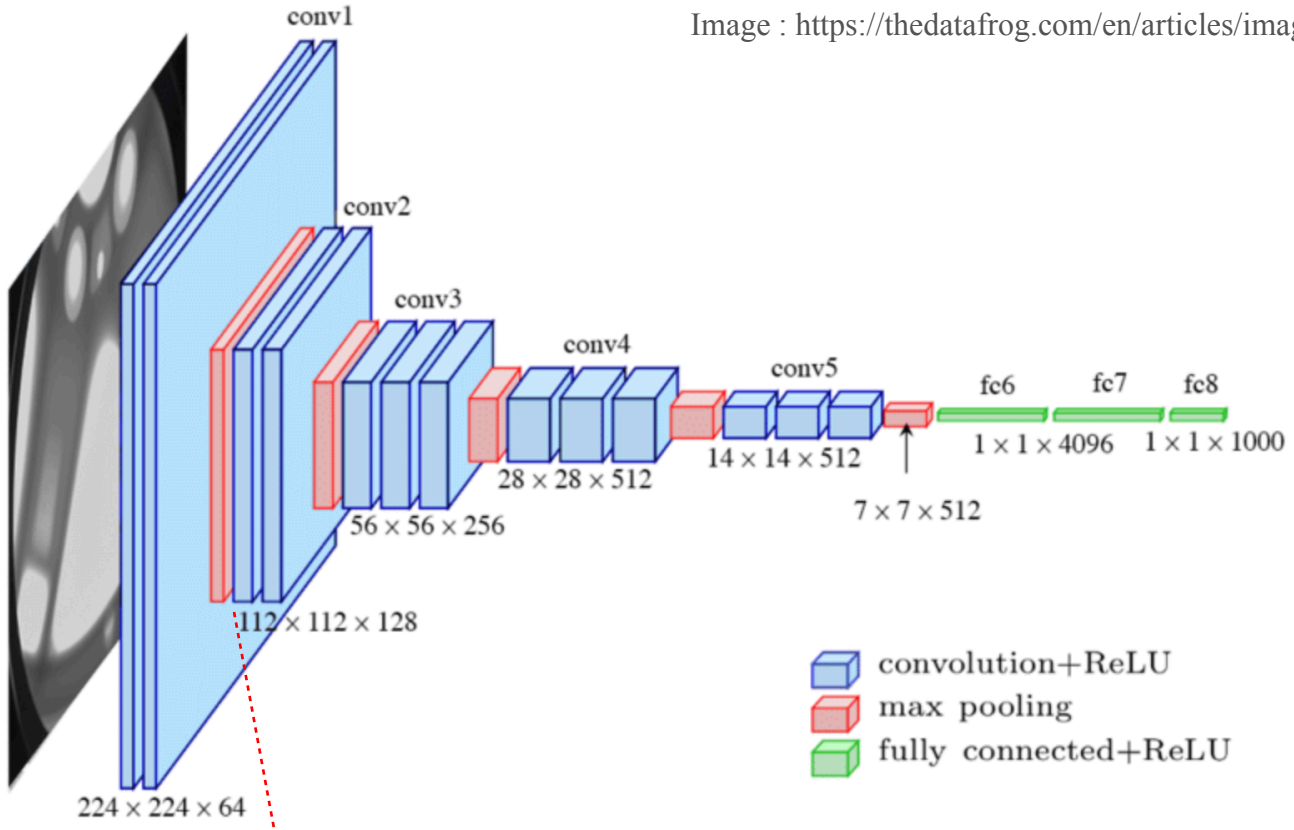


Image : https://thedatafrog.com/en/articles/image-recognition-transfer-learning/



Convolution de [64 canaux de dimension 112×112] vers [128 canaux de dimension 112×112] pour l'observation b du mini-batch

$$\left(o^{l}_{c,f,g}\right)^{b} \quad \begin{array}{l} c = 1,\ldots,64 \\ f = 1,\ldots,112 \\ g = 1,\ldots,112 \\ b = 1,\ldots,B \end{array}$$

$$\left(o^{l+1}_{c,f,g}\right)^{b} \quad \begin{array}{l} c = 1,\ldots,128 \\ f = 1,\ldots,112 \\ g = 1,\ldots,112 \\ b = 1,\ldots,B \end{array}$$

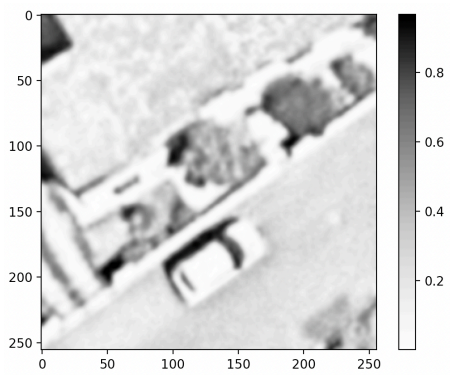
$$\left(s^{l+1}_{c,f,g}\right)^{b} = bias^{l}_{c} + \sum_{c'=1}^{64} \sum_{v=-V}^{V} \sum_{w=-W}^{W} K^{l}_{c',c,v,w} \left(o^{l}_{c',f+v,g+w}\right)^{b}$$

$$\left(o^{l+1}_{c,f,g}\right)^{b} = \varphi\left(\left(s^{l+1}_{c,f,g}\right)^{b}\right) \qquad \forall c = 1,\ldots,128$$

2) Calcul d'un gradient dans un réseau de neurones

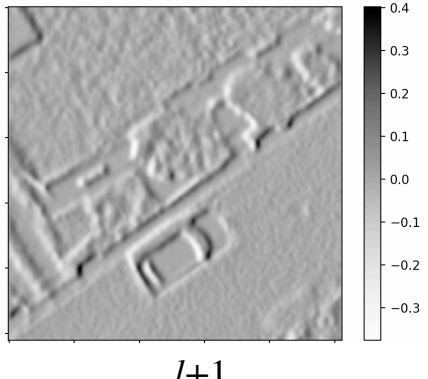
TP2

On se restreint à la convolution d'un canal à un canal…
… mais on va optimiser en CUDA le code la backpropagation d'une couche convolutive !



$o^l_{m,n}$

$$K^l = \frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$



$o^{l+1}_{m,n}$

Modèle *forward* :

$$\begin{cases} s^{l+1}_{f,g} = \sum_{v=-V}^{V} \sum_{w=-W}^{W} K^l_{v,w} o^l_{f+v,g+w} \\ o^{l+1}_{f,g} = \varphi(s^{l+1}_{f,g}) \end{cases}$$

Equations *backward* :

$$\frac{\partial R^i_\theta}{\partial o^l_{m,n}} = \sum_{v=-V}^{V} \sum_{w=-W}^{W} \frac{\partial R^i_\theta}{\partial o^{l+1}_{m+v,n+w}} \frac{\partial o^{l+1}_{m+v,n+w}}{\partial s^{l+1}_{m+v,n+w}} \frac{\partial s^{l+1}_{m+v,n+w}}{\partial o^l_{m,n}} = \frac{\partial R^i_\theta}{\partial o^{l+1}_{m+v,n+w}} \varphi'(s^{l+1}_{m+v,n+w}) K^l_{-v,-w}$$

$$\frac{\partial R^i_\theta}{\partial K^l_{v,w}} = \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{\partial R^i_\theta}{\partial o^{l+1}_{m,n}} \frac{\partial o^{l+1}_{m,n}}{\partial s^{l+1}_{m,n}} \frac{\partial s^{l+1}_{m,n}}{\partial K^l_{v,w}} = \frac{\partial R^i_\theta}{\partial o^{l+1}_{m,n}} \varphi'(s^{l+1}_{m,n}) o^l_{m+v,n+w}$$