DEVLOG RÉSEAU DU DEVELOPPEMENT LOGICIEL Recherche et Enseignement Supérieur

cnrs dépasser les frontières www.cnrs.fr

INSTITUT de MATHEMATIQUES de TOULOUSE

ANITI ARTIFICIAL & NATURAL INTELLIGENCE TOULOUSE INSTITUTE

cimi Centre International de Mathématiques et d'Informatique TOULOUSE

Introduction au Deep Learning avec PyTorch

Partie 2 : Réseaux de neurones

Laurent Risser

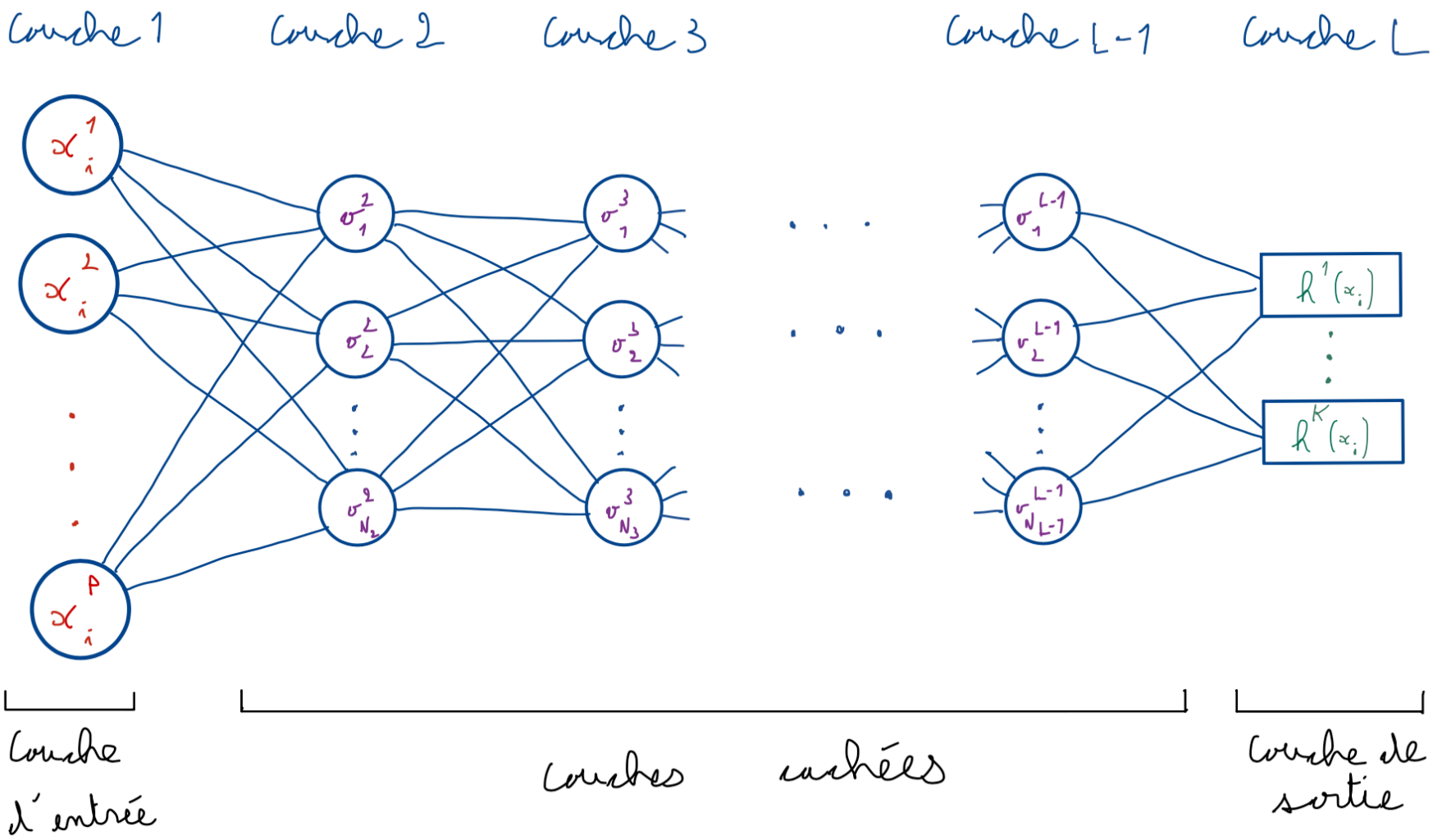
Ingénieur de Recherche à l'Institut de Mathématiques de Toulouse et au 3IA ANITI

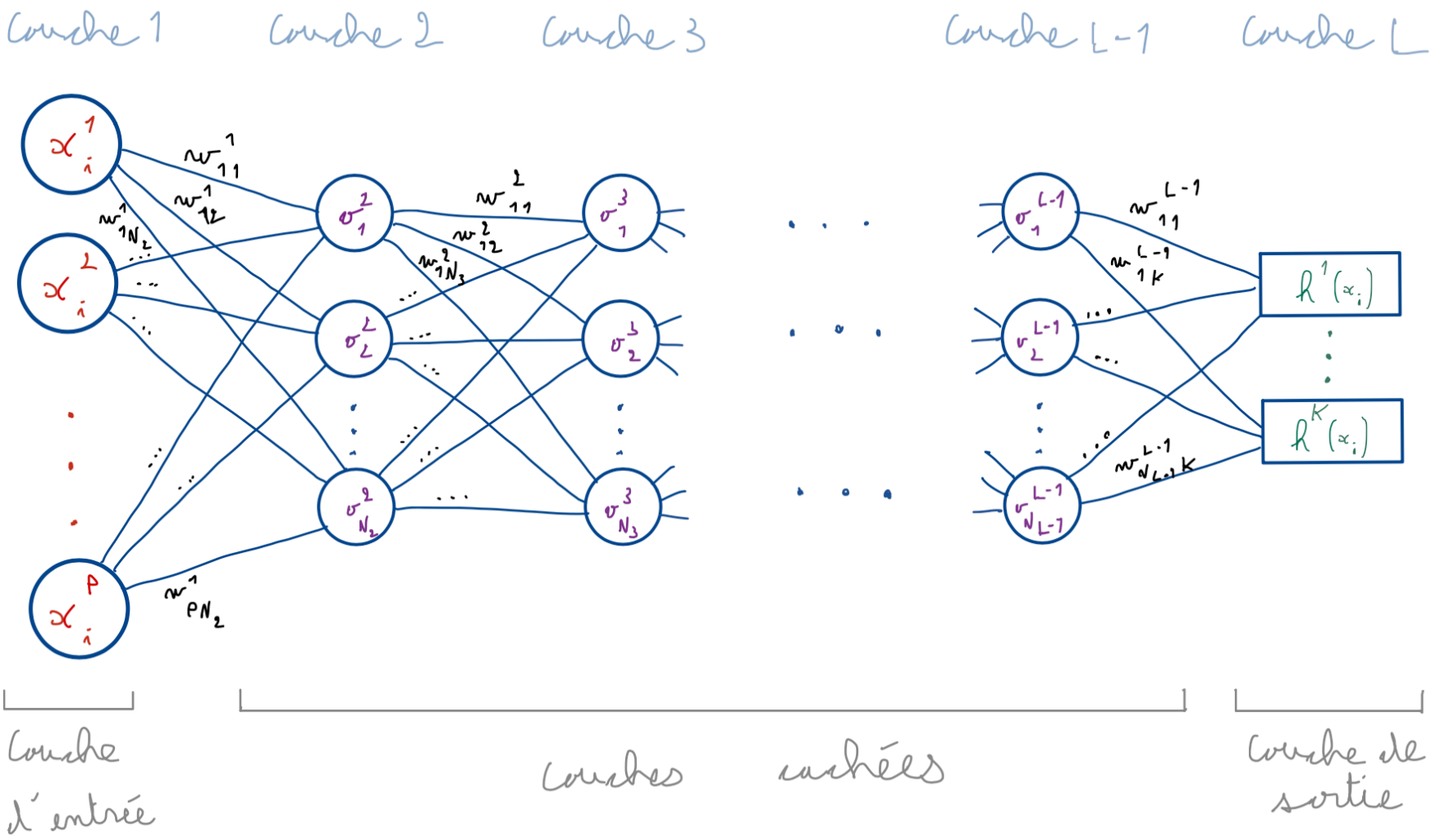
lrisser@math.univ-toulouse.fr

Nous avons vu le modèle linéaire pour faire des prédictions :

- **Modèle linéaire :** $\widehat{y_{new}} = h_\Theta(x_{new}) = w_0 + \sum_{j=1}^{p} w_j x_{new}^j$
- **Paramètres :** $\Theta = \{w_0, w_1, \ldots, w_p\}$

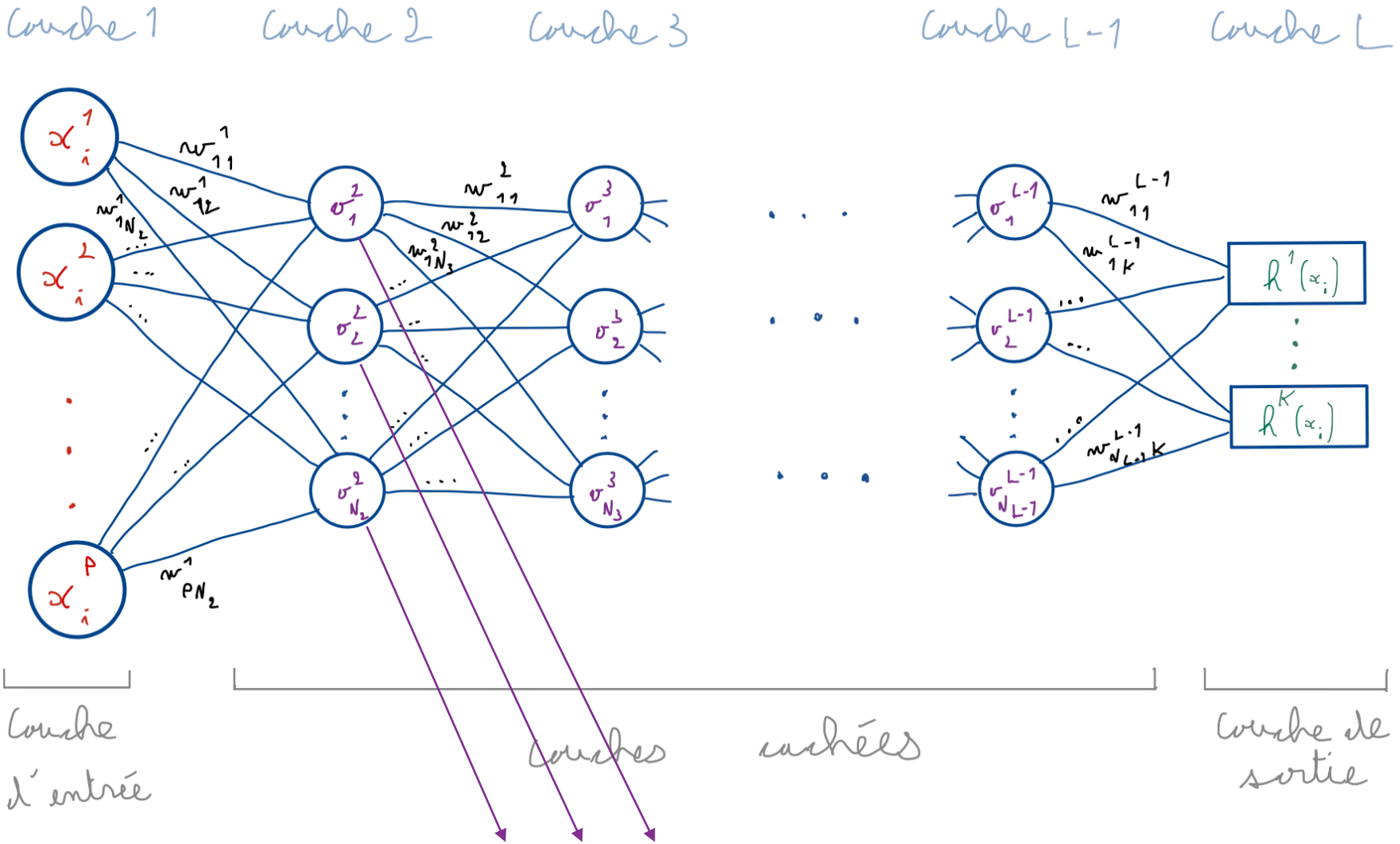
Généralisons le avec une architecture de réseau de neurone classique : **Le perceptron multi-couches**


Couche 1
Couche 2
Couche 3
Couche L-1
Couche L
Couche d'entrée
couches cachées
Couche de sortie



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

$$o_k^1 = x_i^k$$

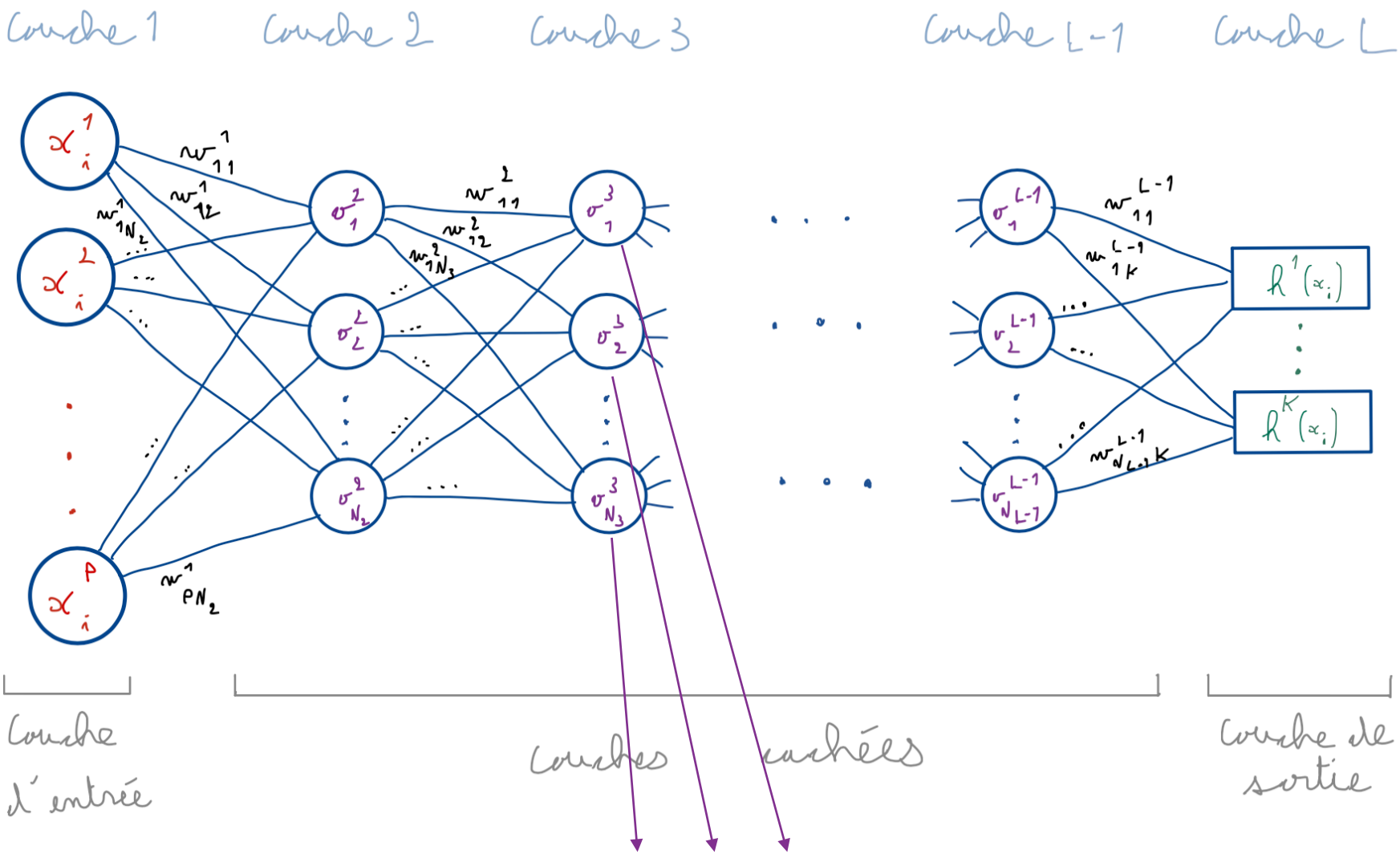


$\forall k = 1,\dots,p$:

$o_k^1 = x_i^k$

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

$\varphi(\,.\,)$ est une fonction linéaire, dite d'activation (ex : ReLU, sigmoïde, …)

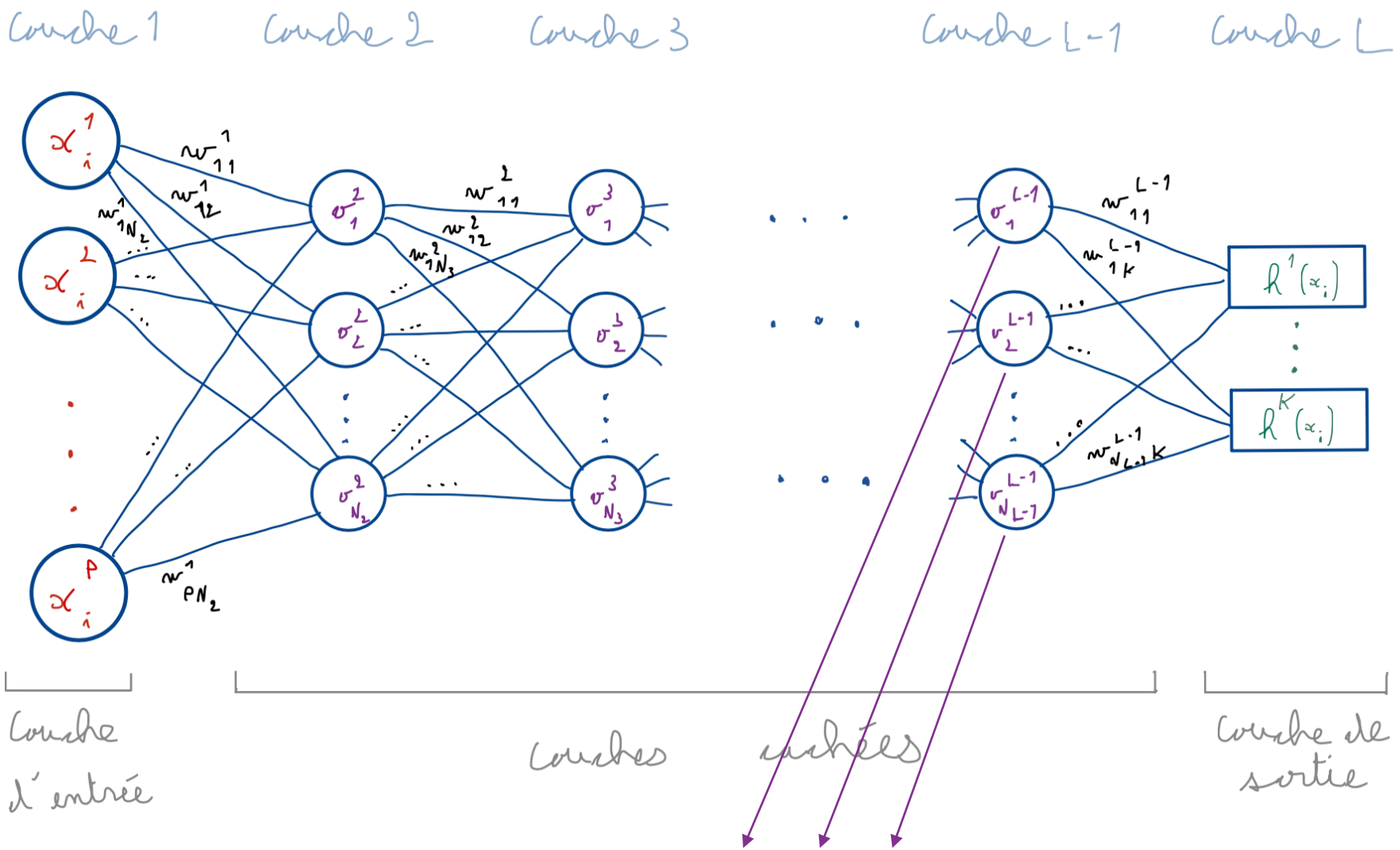


$\forall k = 1,\dots,p$:

$o_k^1 = x_i^k$

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

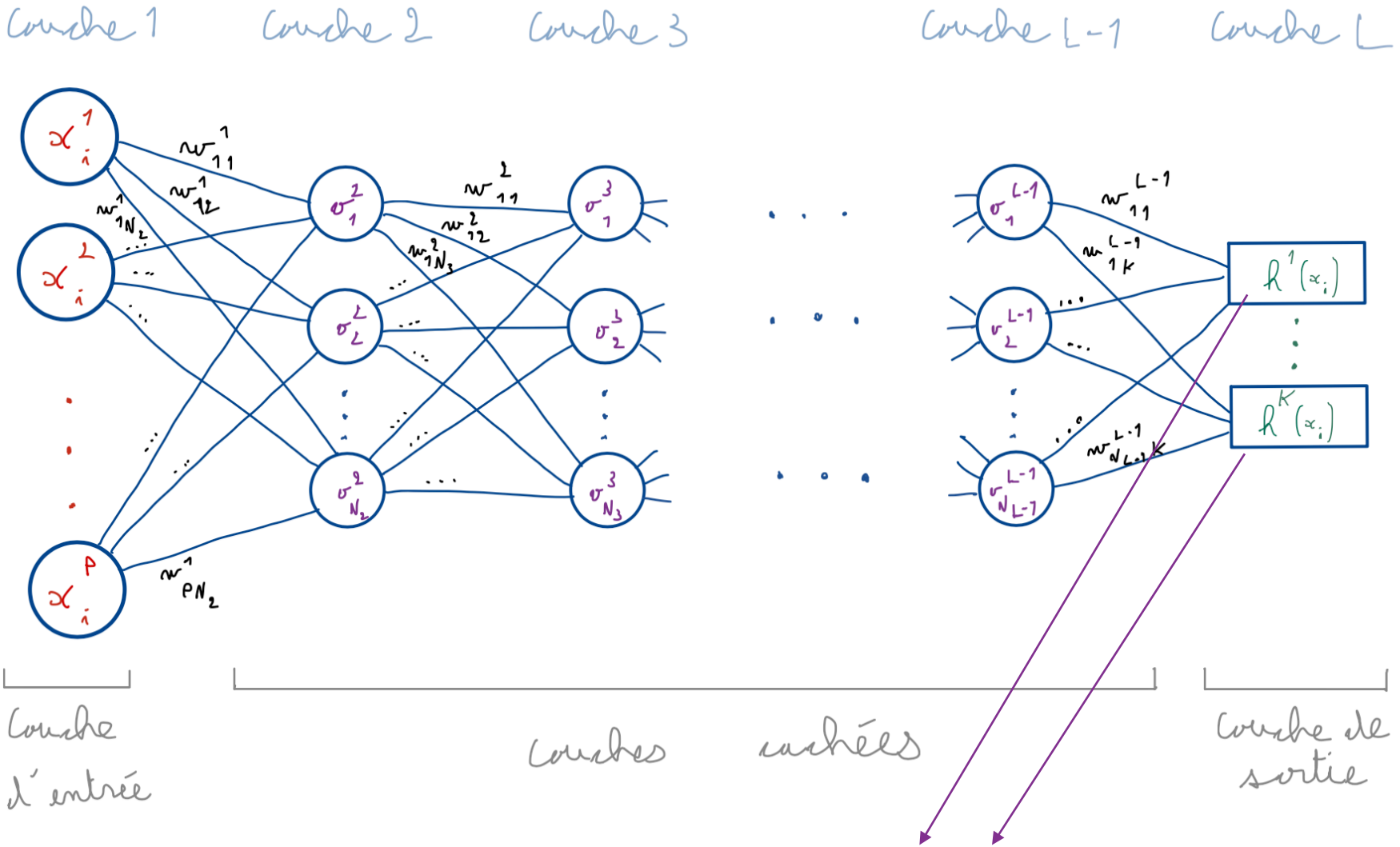
$\varphi(\,.\,)$ est une fonction linéaire, dite d'activation (ex : ReLU, sigmoïde, …)



$$\forall k = 1,\dots,p : \quad o_k^1 = x_i^k$$

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

$\varphi(\,.\,)$ est une fonction linéaire, dite d'activation (ex : ReLU, sigmoïde, …)



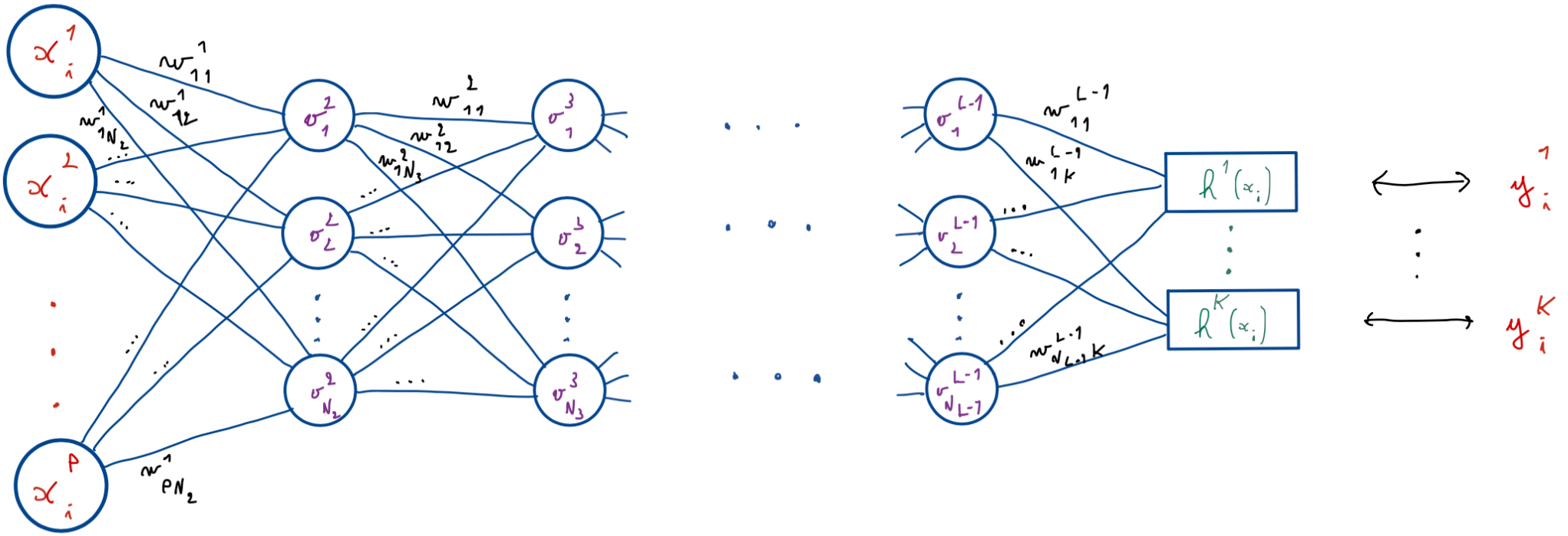
$\forall k = 1,\dots,p$:

$o_k^1 = x_i^k$

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

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

$h^k(x_i) = o_k^L$



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

Exemple d'application : Reconnaissance de caractéristiques dans des images

Jeu de données *CelebA*

Une **entrée** $x_i = (x_i^1, x_i^2, \ldots, x_i^p)$ est **ici** une image mise à plat.

Pour une image RGB de taille 400x300 :

- $(x_i^1, \ldots, x_i^{400})$ est la *1ere ligne* de l'image sur le canal *Rouge*
- $(x_i^{401}, \ldots, x_i^{800})$ est la *2eme ligne* de l'image sur le canal *Rouge*
- …
- $(x_i^{299*400+1}, \ldots, x_i^{300*400})$ est la *300eme ligne* de l'image sur le canal *Rouge*
- $(x_i^{300*400+1}, \ldots, x_i^{300*400+400})$ est la *1ere ligne* de l'image sur le canal *Vert*
- …

Une **sortie** $y_i = (y_i^1, y_i^2, \ldots, y_i^K)$ est **ici** le fait d'observer différentes caractéristiques dans l'image

- $y_i^1 = 1$ si la personne a des lunettes et $y_i^1 = 0$ sinon ;
- Pareil pour y_i^2 avec le fait de sourire;
- …

Exemple d'application : Reconnaissance de caractéristiques dans des images

Jeu de données *CelebA*

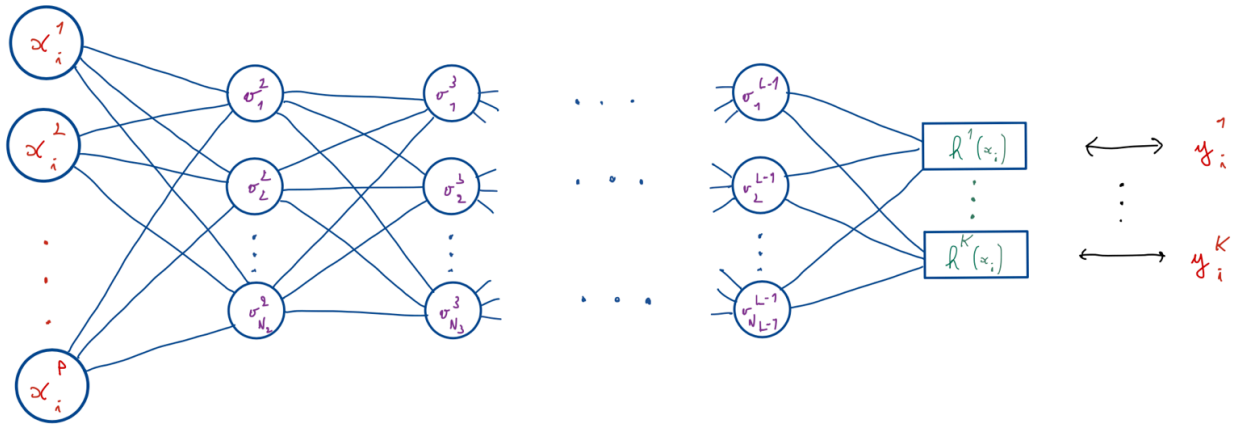
Une **entrée** $x_i = (x_i^1, x_i^2, \ldots, x_i^p)$ est **ici** une image mise à plat.

Pour une image RGB de taille 400x300 :

- $(x_i^1, \ldots, x_i^{400})$ est la *1ere ligne* de l'image sur le canal *Rouge*
- $(x_i^{401}, \ldots, x_i^{800})$ est la *2eme ligne* de l'image sur le canal *Rouge*
- …
- $(x_i^{299*400+1}, \ldots, x_i^{300*400})$ est la *300eme ligne* de l'image sur le canal *Rouge*
- $(x_i^{300*400+1}, \ldots, x_i^{300*400+400})$ est la *1ere ligne* de l'image sur le canal *Vert*
- …

Une **sortie** $y_i = (y_i^1, y_i^2, \ldots, y_i^K)$ est **ici** le fait d'observer différentes caractéristiques dans l'image

- $y_i^1 = 1$ si la personne a des lunettes et $y_i^1 = 0$ sinon ;
- Pareil pour y_i^2 avec le fait de sourire;
- …



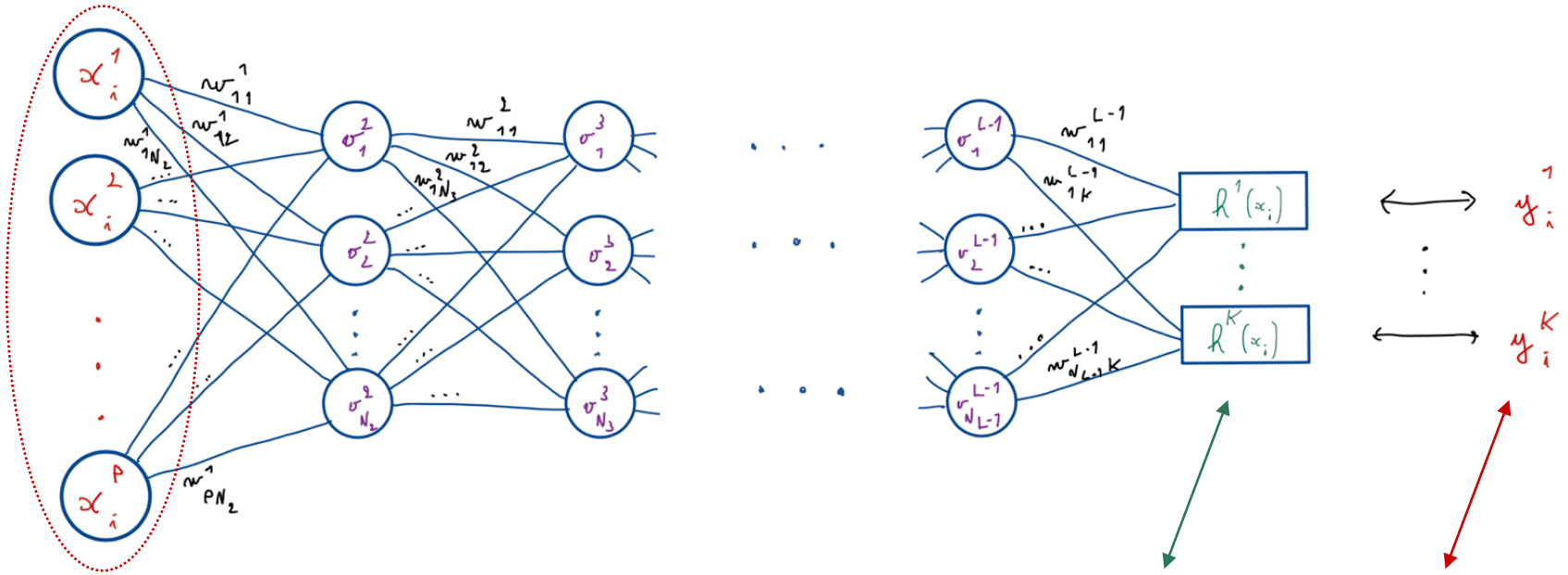
Un *loss* raisonnable ici est :

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

(Mais il en existe de nombreux autres… qui doivent cependant rester dérivables)

Exemple d'application : Reconnaissance de caractéristiques dans des images

Avec des poids $\Theta = \{w_{pk}^l\}_{p,k,l}$ **bien** calibrés :

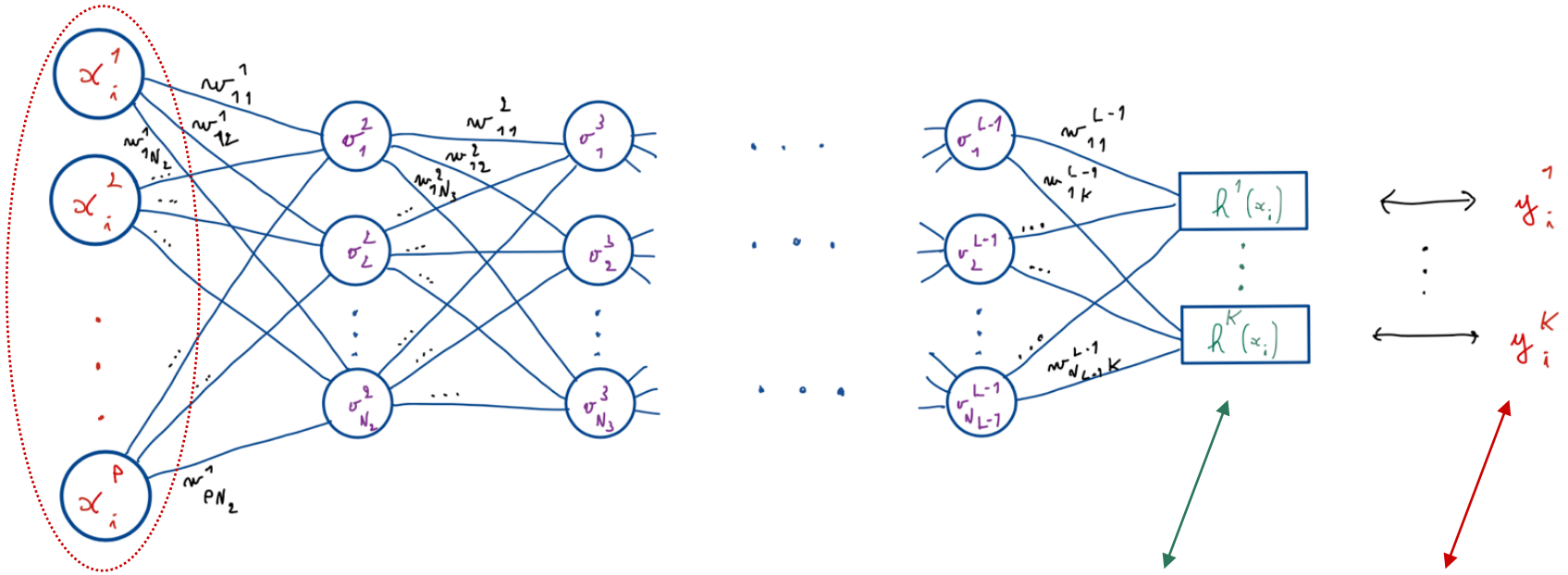


$$h_\Theta^1(x_i) = 0.001 \quad \text{et} \quad y_i^1 = 0$$
$$h_\Theta^1(x_i) = 0.003 \quad \text{et} \quad y_i^2 = 0$$
$$\dots$$

$$loss\left(h_\Theta(x_i), y_i\right) = \sum_{k=1}^{K} (h_\Theta^k(x_i) - y_i^k)^2 = 0.0004$$

Exemple d'application : Reconnaissance de caractéristiques dans des images

Avec des poids $\Theta = \{w_{pk}^l\}_{p,k,l}$ **bien** calibrés :

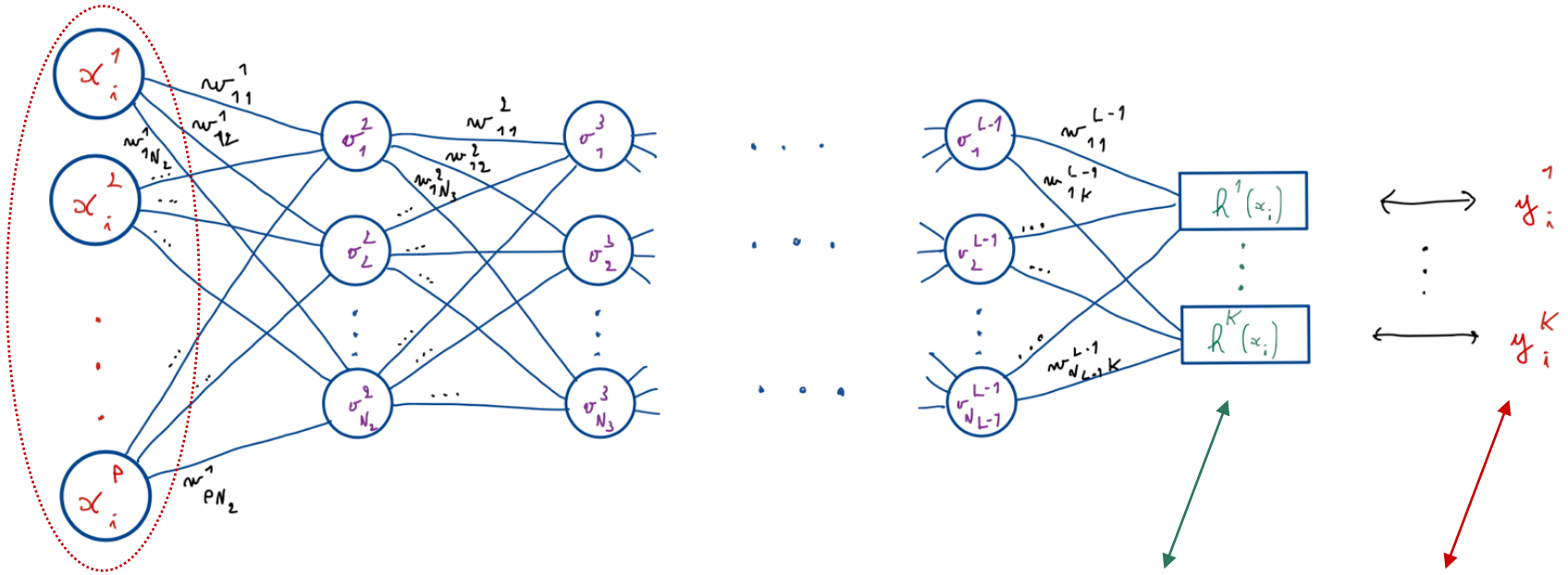


$$h_\Theta^1(x_i) = 0.999 \quad \text{et} \quad y_i^1 = 1$$
$$h_\Theta^1(x_i) = 0.980 \quad \text{et} \quad y_i^2 = 1$$
$$\dots$$

$$loss\left(h_\Theta(x_i), y_i\right) = \sum_{k=1}^{K} (h_\Theta^k(x_i) - y_i^k)^2 = 0.0031$$

Exemple d'application : Reconnaissance de caractéristiques dans des images

Avec des poids $\Theta = \{w_{pk}^l\}_{p,k,l}$ **mal** calibrés :

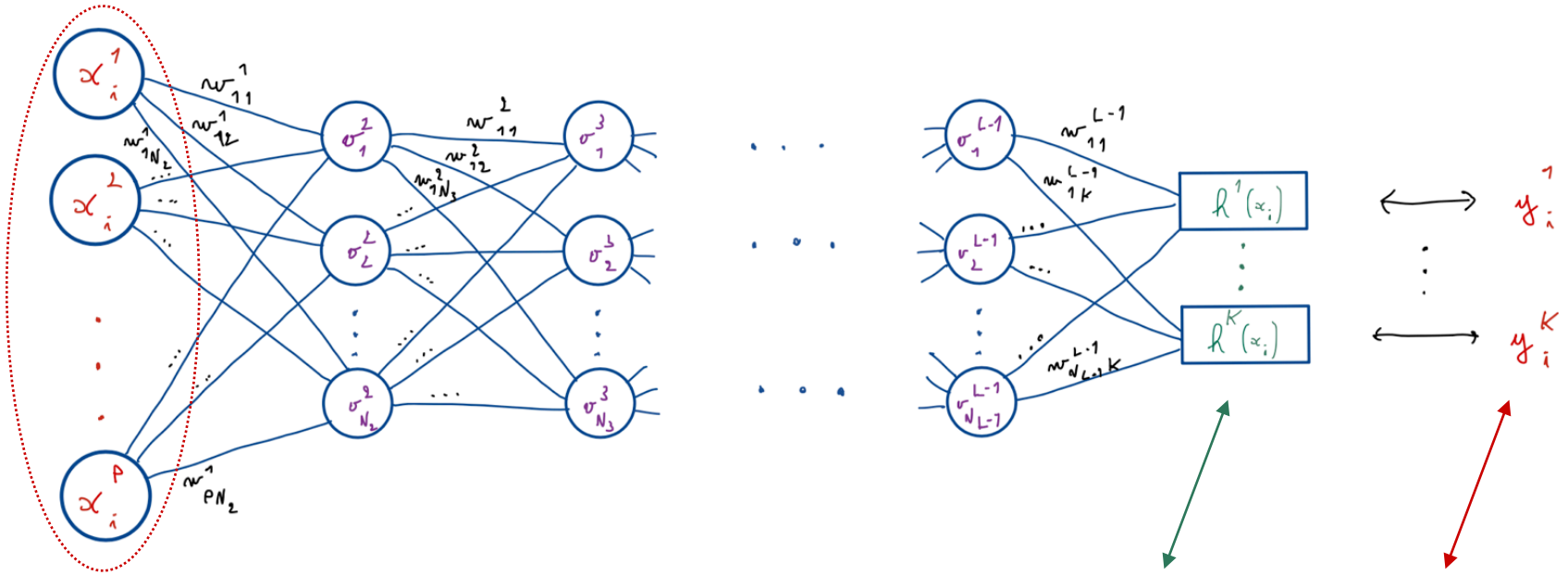


$$h_\Theta^1(x_i) = 0.402 \quad \text{et} \quad y_i^1 = 0$$
$$h_\Theta^1(x_i) = 0.831 \quad \text{et} \quad y_i^2 = 0$$
$$\dots$$

$$loss\left(h_\Theta(x_i), y_i\right) = \sum_{k=1}^{K} (h_\Theta^k(x_i) - y_i^k)^2 = 0.376$$

Exemple d'application : Reconnaissance de caractéristiques dans des images

Avec des poids $\Theta = \{w_{pk}^l\}_{p,k,l}$ **mal** calibrés :

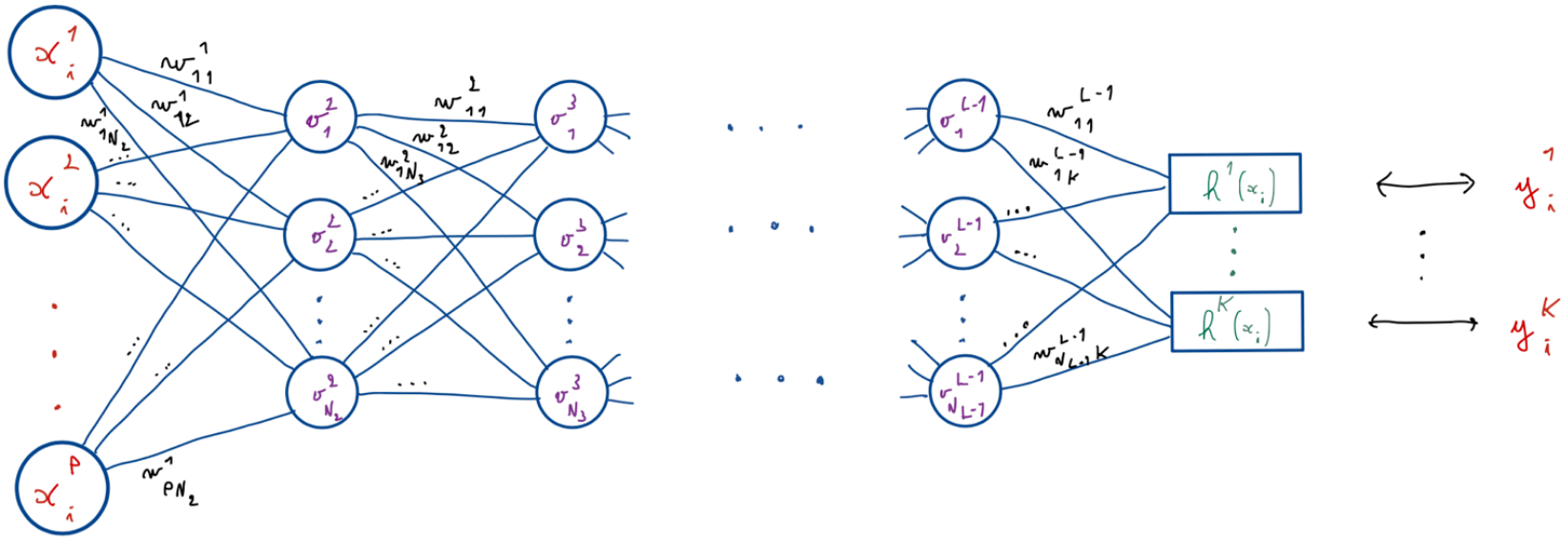


$h_\Theta^1(x_i) = 0.823$ et $y_i^1 = 1$

$h_\Theta^1(x_i) = 0.789$ et $y_i^2 = 1$

...

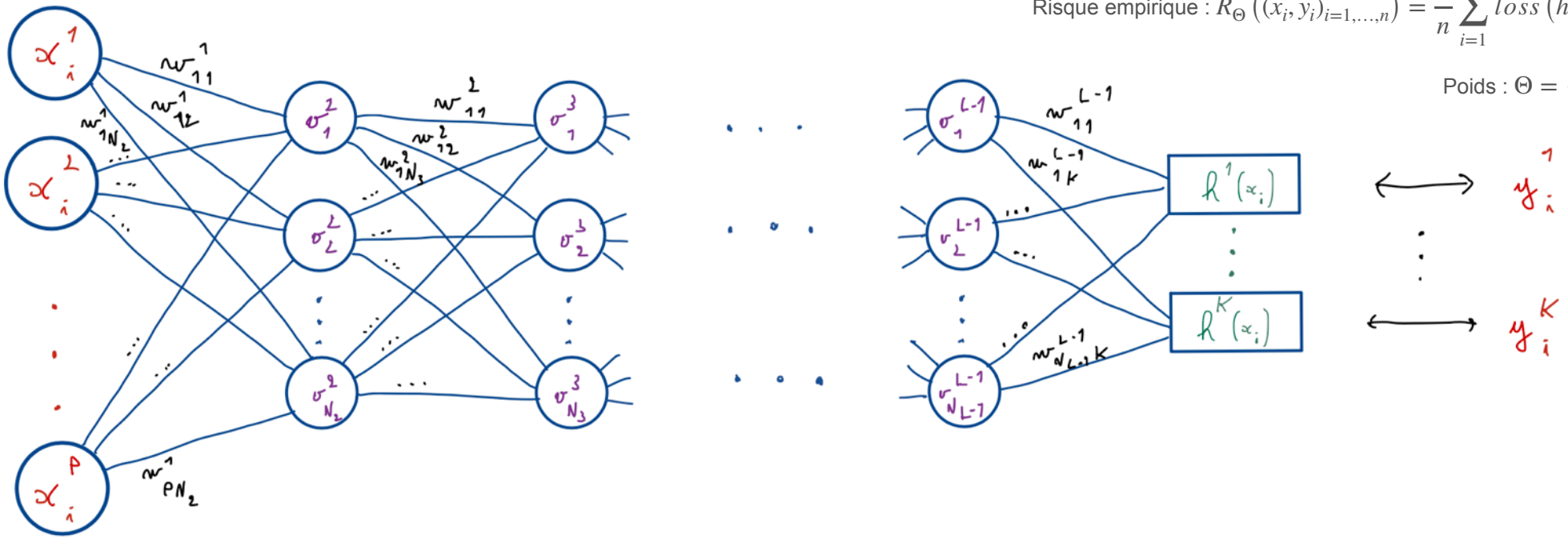
$$loss\left(h_\Theta(x_i), y_i\right) = \sum_{k=1}^{K} (h_\Theta^k(x_i) - y_i^k)^2 = 0.187$$



Recherche des poids optimaux pour minimiser en moyenne le loss sur un jeu d'apprentissage (*i.e.* le risque empirique) →**descente de gradient en grande dimension**

Risque empirique : $R_\Theta\left((x_i,y_i)_{i=1,\dots,n}\right) = \frac{1}{n}\sum_{i=1}^{n} loss\left(h_\Theta(x_i),y_i\right)$

Poids : $\Theta = \{w_{pk}^l\}_{p,k,l}$



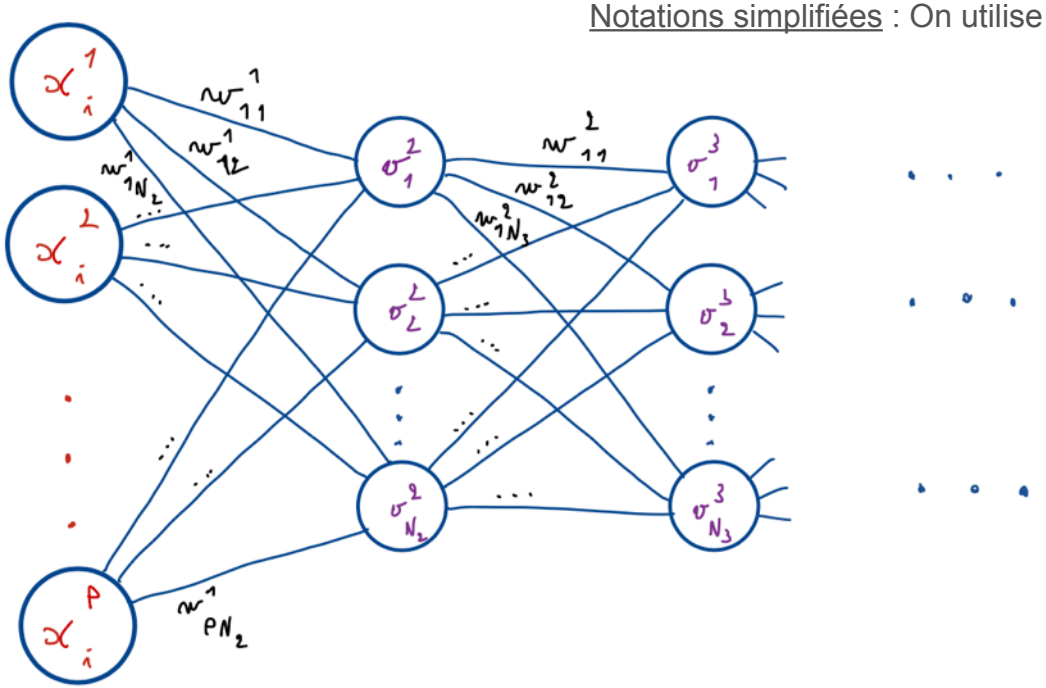
Gradient du risque empirique :

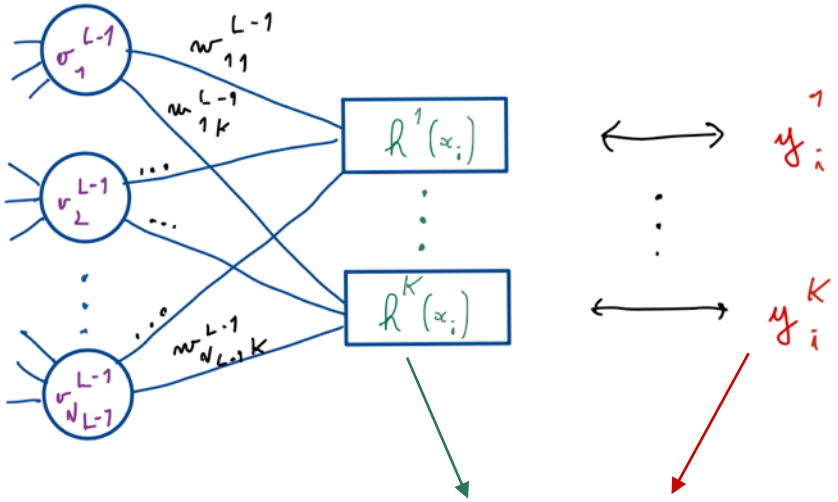
$$\nabla R_\Theta\left((x_i,y_i)_{i=1,\dots,n}\right) = \left(\frac{\partial R_\Theta(\dots)}{\partial w_{1,1}^1}, \frac{\partial R_\Theta(\dots)}{\partial w_{1,2}^1}, \dots, \frac{\partial R_\Theta(\dots)}{\partial w_{N_{L-1},K}^{N_{L-1}}}\right)^\intercal$$

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

Calculable analytiquement… mais subtile à calculer car les $h^k(x_i)$ dépendent de beaucoup de fonctions dépendant des $\{w_{pk}^l\}_{p,k,l}$ imbriquées les unes dans les autres

Notations simplifiées : On utilise ici E pour désigner $R_\Theta(\dots)$ et on se focalise sur un seul $k \in \{1,\dots,K\}$

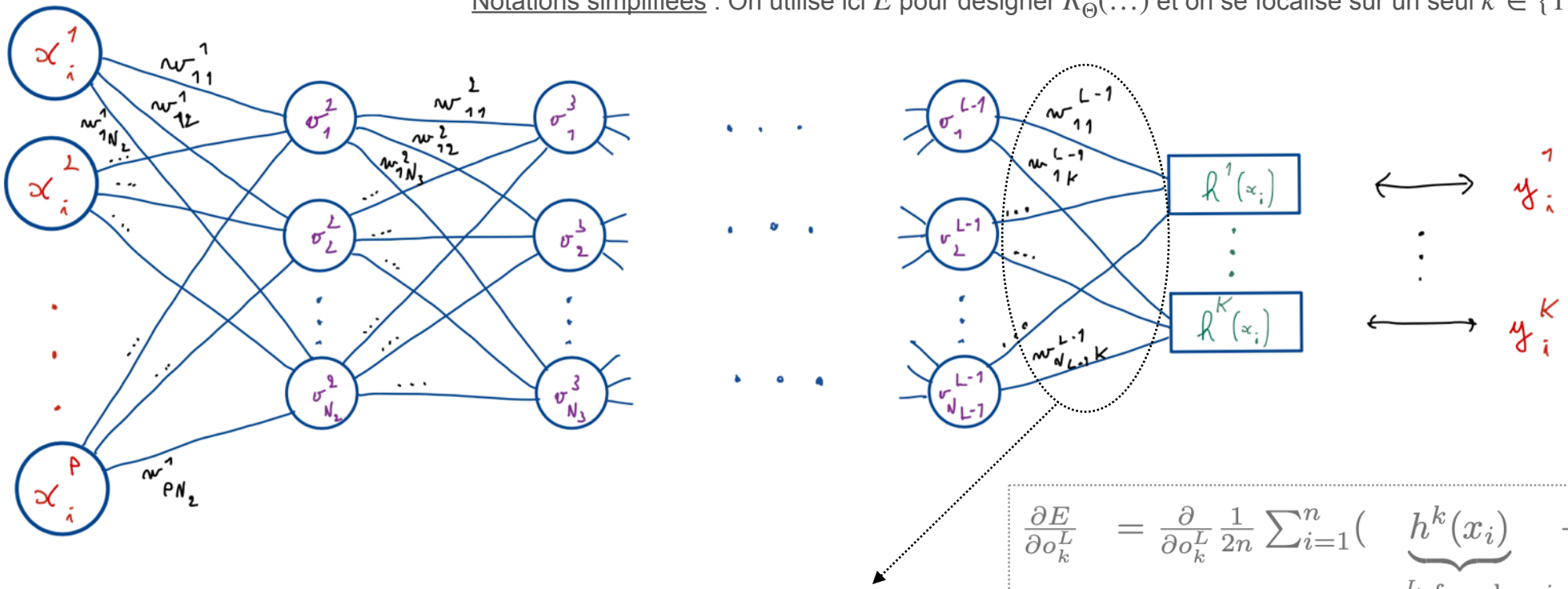




$$\frac{\partial E}{\partial o_k^L} = \frac{\partial}{\partial o_k^L} \frac{1}{2n} \sum_{i=1}^n (\underbrace{h^k(x_i)}_{=o_k^L \text{ for obs. } i} - y_i^k)^2$$

$$= \frac{1}{n} \sum_{i=1}^n \left(h^k(x_i) - y_i^k \right)$$

Notations simplifiées : On utilise ici E pour désigner $R_\Theta(\dots)$ et on se focalise sur un seul $k \in \{1,\dots,K\}$



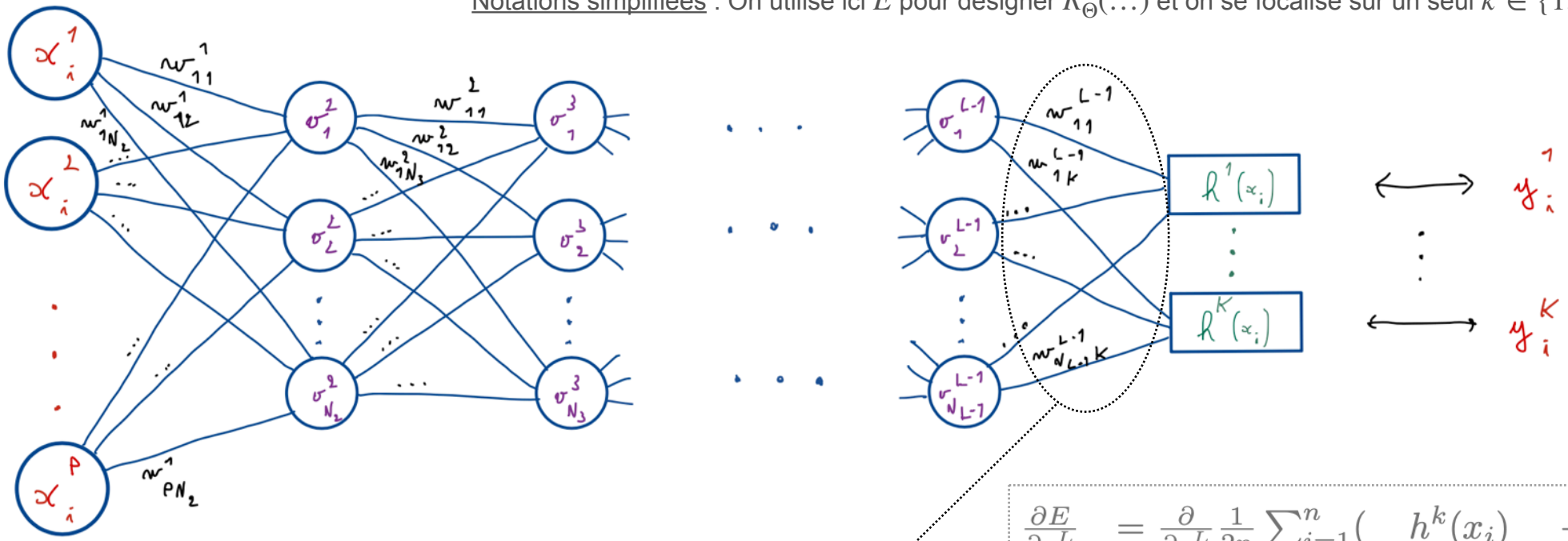
$$\frac{\partial E}{\partial w_{pk}^l} = \frac{\partial E}{\partial o_k^{l+1}} \frac{\partial o_k^{l+1}}{\partial s_k^{l+1}} \frac{\partial s_k^{l+1}}{\partial w_{pk}^l}$$

$l = L-1$

$$\frac{\partial E}{\partial o_k^L} = \frac{\partial}{\partial o_k^L} \frac{1}{2n} \sum_{i=1}^n (\underbrace{h^k(x_i)}_{=o_k^L \text{ for obs. } i} - y_i^k)^2$$

$$= \frac{1}{n} \sum_{i=1}^n \left(h^k(x_i) - y_i^k \right)$$

Notations simplifiées : On utilise ici E pour désigner $R_\Theta(\dots)$ et on se focalise sur un seul $k \in \{1,\dots,K\}$



$$\frac{\partial E}{\partial o_k^L} = \frac{\partial}{\partial o_k^L}\frac{1}{2n}\sum_{i=1}^n(\underbrace{h^k(x_i)}_{=o_k^L \text{ for obs. } i} - y_i^k)^2$$

$$= \frac{1}{n}\sum_{i=1}^n \left(h^k(x_i) - y_i^k\right)$$

$$\frac{\partial E}{\partial w_{pk}^l} = \frac{\partial E}{\partial o_k^{l+1}}\frac{\partial o_k^{l+1}}{\partial s_k^{l+1}}\frac{\partial s_k^{l+1}}{\partial w_{pk}^l}$$

$l = L-1$

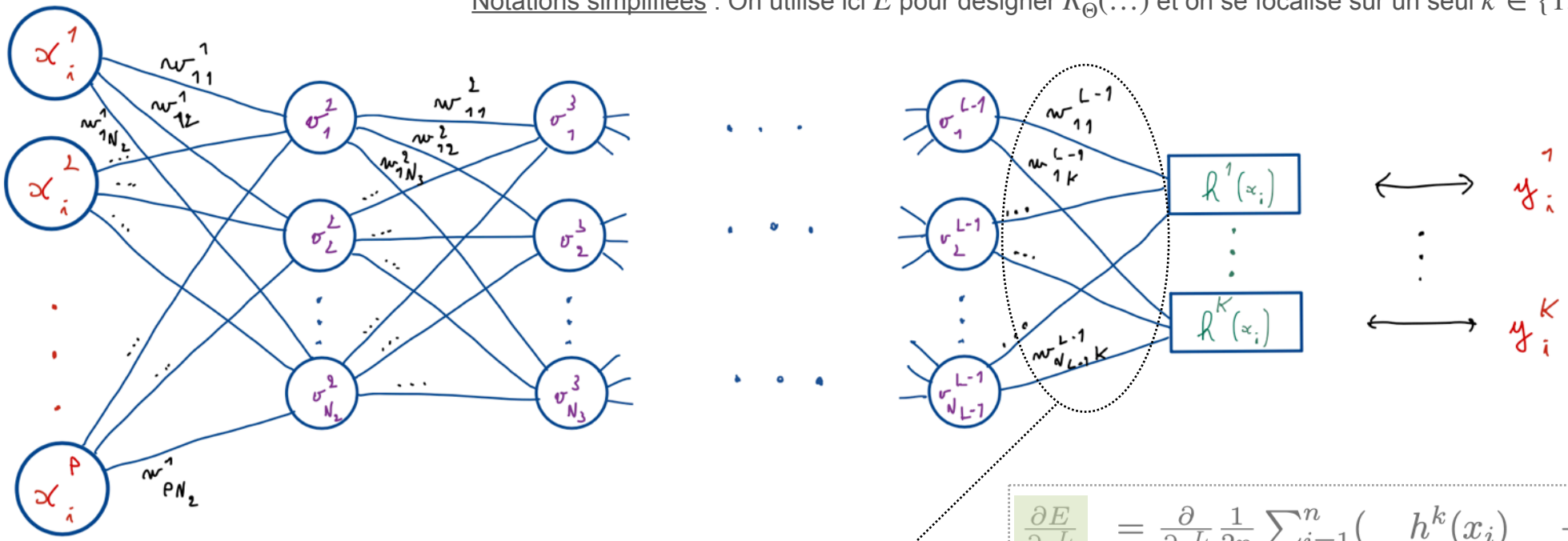
$$\frac{\partial o_k^{l+1}}{\partial s_k^{l+1}} = \frac{\partial}{\partial s_k^{l+1}}\varphi(s_k^{l+1}) \text{ [connu]}$$

$l = L-1$

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

$l = L-1$

Notations simplifiées : On utilise ici E pour désigner $R_\Theta(\dots)$ et on se focalise sur un seul $k \in \{1,\dots,K\}$



$$\frac{\partial E}{\partial o_k^L} = \frac{\partial}{\partial o_k^L} \frac{1}{2n} \sum_{i=1}^n \Big(\underbrace{h^k(x_i)}_{=o_k^L \text{ for obs. } i} - y_i^k \Big)^2$$

$$= \frac{1}{n} \sum_{i=1}^n \left(h^k(x_i) - y_i^k \right)$$

$$\frac{\partial E}{\partial w_{pk}^l} = \frac{\partial E}{\partial o_k^{l+1}} \frac{\partial o_k^{l+1}}{\partial s_k^{l+1}} \frac{\partial s_k^{l+1}}{\partial w_{pk}^l}$$

$l = L-1$

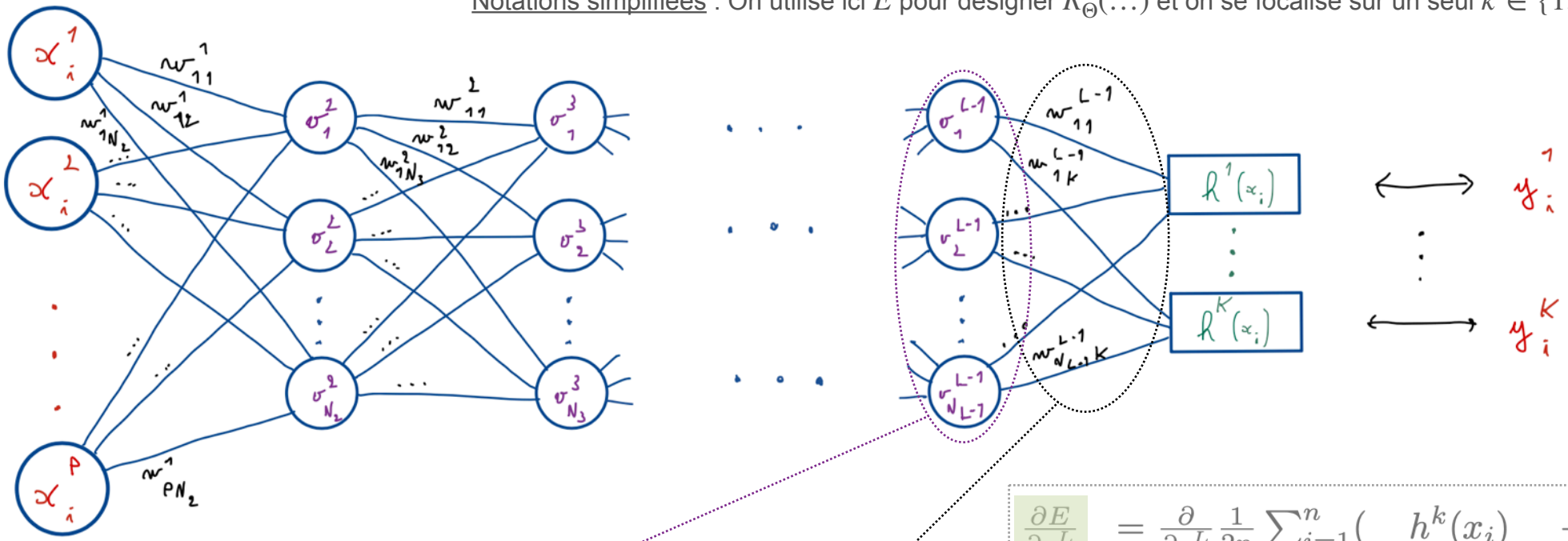
$$\frac{\partial o_k^{l+1}}{\partial s_k^{l+1}} = \frac{\partial}{\partial s_k^{l+1}} \varphi(s_k^{l+1}) \text{ [connu]}$$

$l = L-1$

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

$l = L-1$

Notations simplifiées : On utilise ici E pour désigner $R_\Theta(\dots)$ et on se focalise sur un seul $k \in \{1,\dots,K\}$



$$\frac{\partial E}{\partial o_k^L} = \frac{\partial}{\partial o_k^L}\frac{1}{2n}\sum_{i=1}^n\big(\underbrace{h^k(x_i)}_{=o_k^L \text{ for obs. } i} - y_i^k\big)^2$$

$$= \frac{1}{n}\sum_{i=1}^n \big(h^k(x_i) - y_i^k\big)$$

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

$l = L-1$

$$\frac{\partial E}{\partial w_{pk}^l} = \frac{\partial E}{\partial o_k^{l+1}}\frac{\partial o_k^{l+1}}{\partial s_k^{l+1}}\frac{\partial s_k^{l+1}}{\partial w_{pk}^l}$$

$l = L-1$

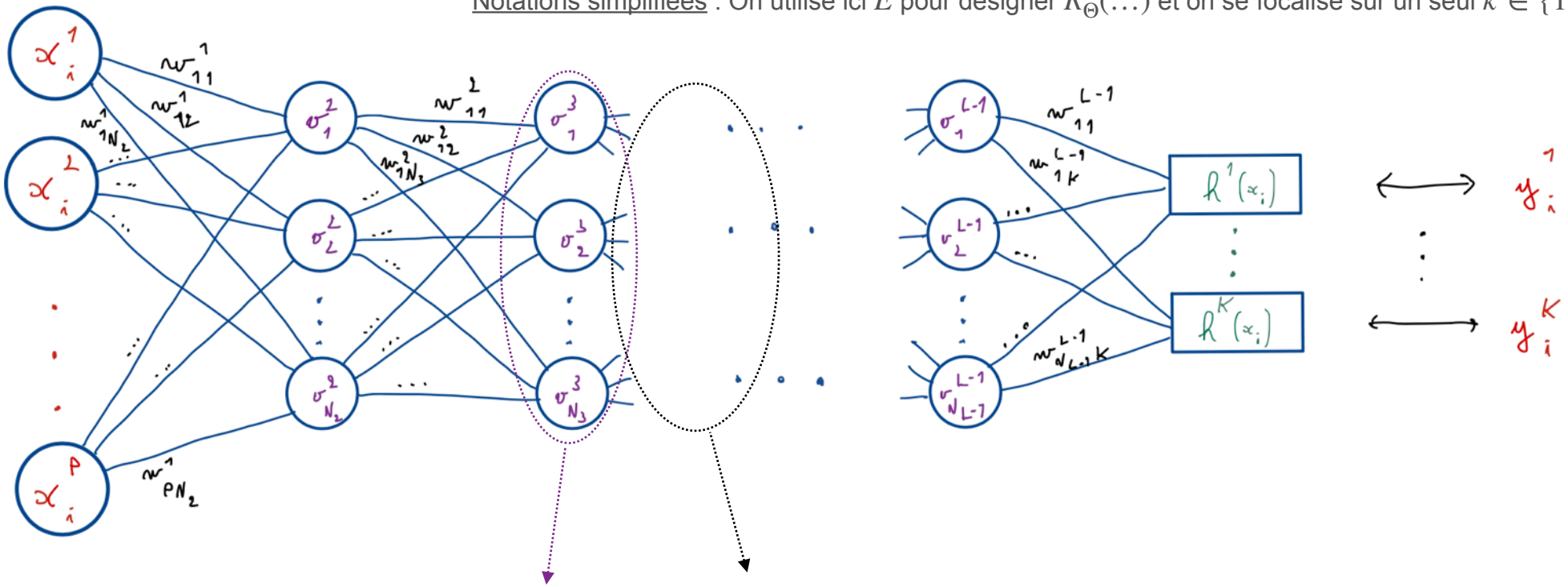
$$\frac{\partial o_k^{l+1}}{\partial s_k^{l+1}} = \frac{\partial}{\partial s_k^{l+1}}\varphi(s_k^{l+1}) \text{ [connu]}$$

$l = L-1$

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

$l = L-1$

Notations simplifiées : On utilise ici E pour désigner $R_\Theta(\dots)$ et on se focalise sur un seul $k \in \{1,\dots,K\}$



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

$l = 3$

$$\frac{\partial E}{\partial w_{pk}^l} = \frac{\partial E}{\partial o_k^{l+1}} \frac{\partial o_k^{l+1}}{\partial s_k^{l+1}} \frac{\partial s_k^{l+1}}{\partial w_{pk}^l}$$

$l = 3$

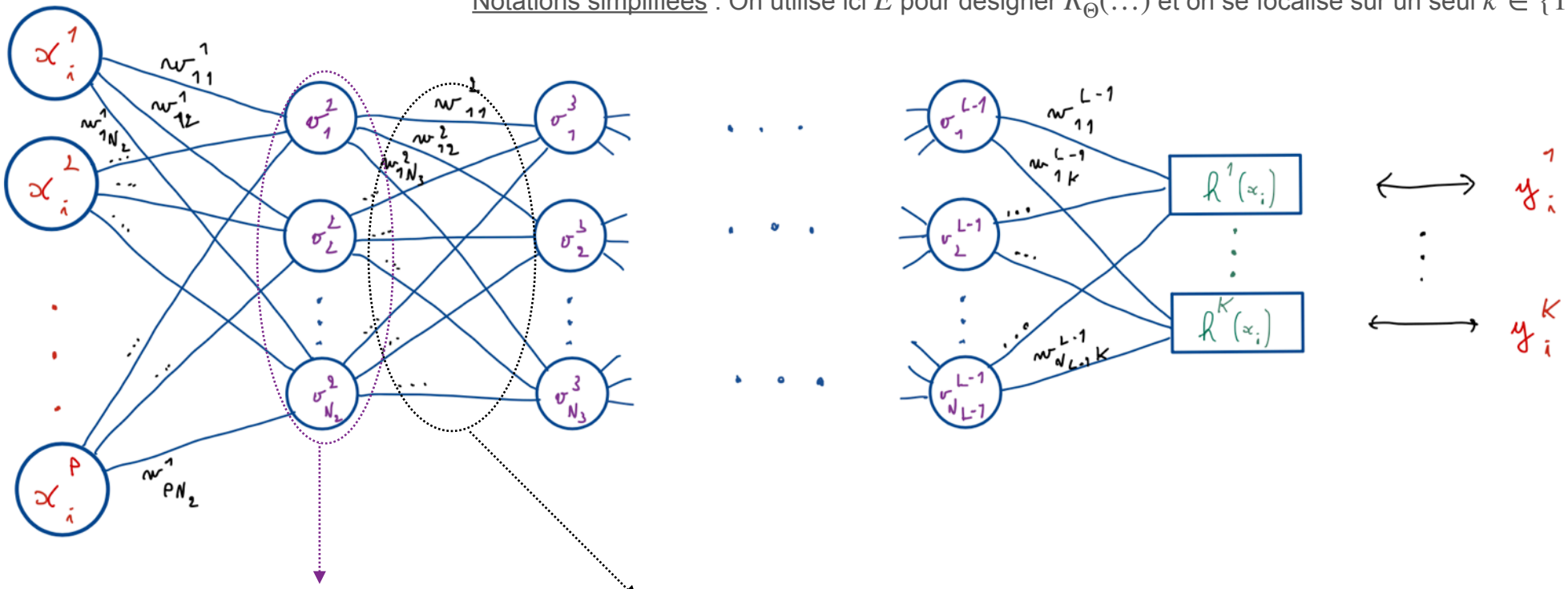
$$\frac{\partial o_k^{l+1}}{\partial s_k^{l+1}} = \frac{\partial}{\partial s_k^{l+1}} \varphi(s_k^{l+1}) \text{ [connu]}$$

$l = 3$

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

$l = 3$

Notations simplifiées : On utilise ici E pour désigner $R_\Theta(\dots)$ et on se focalise sur un seul $k \in \{1,\dots,K\}$



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

$l = 2$

$$\frac{\partial E}{\partial w_{pk}^l} = \frac{\partial E}{\partial o_k^{l+1}} \frac{\partial o_k^{l+1}}{\partial s_k^{l+1}} \frac{\partial s_k^{l+1}}{\partial w_{pk}^l}$$

$l = 2$

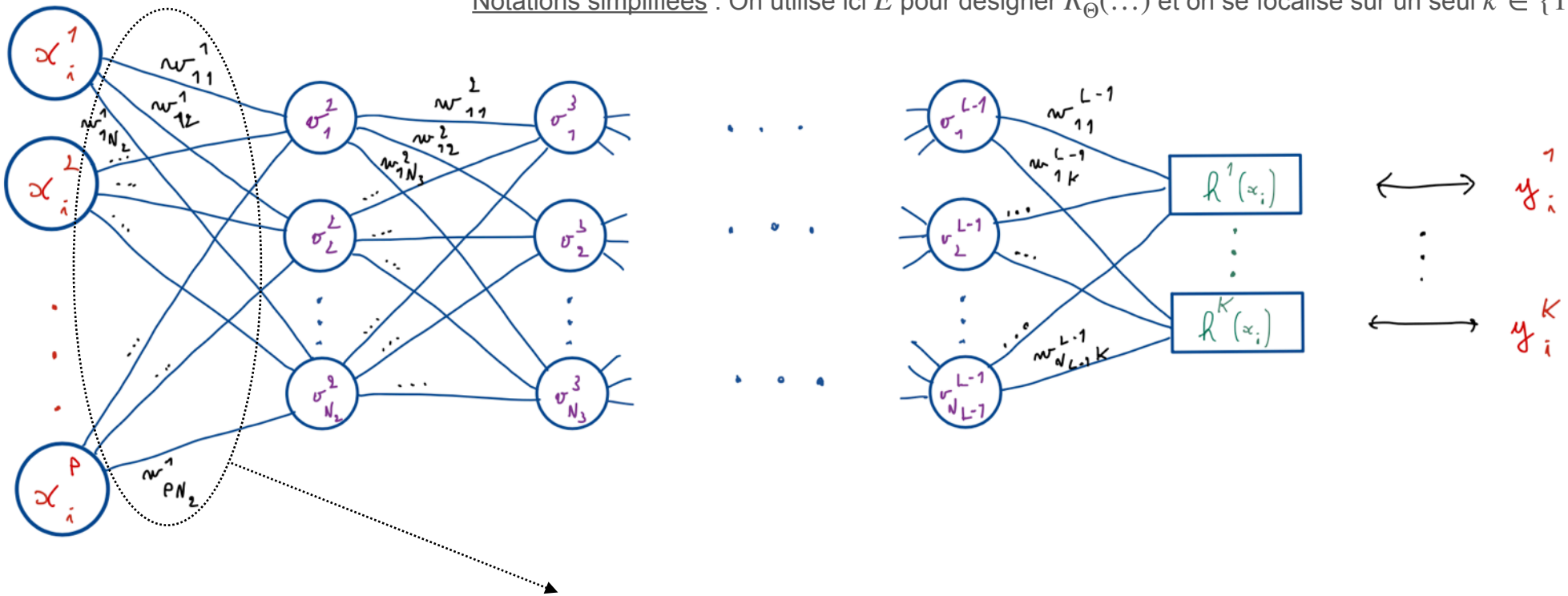
$$\frac{\partial o_k^{l+1}}{\partial s_k^{l+1}} = \frac{\partial}{\partial s_k^{l+1}} \varphi(s_k^{l+1}) \text{ [connu]}$$

$l = 2$

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

$l = 2$

Notations simplifiées : On utilise ici E pour désigner $R_\Theta(\dots)$ et on se focalise sur un seul $k \in \{1,\dots,K\}$



$$\frac{\partial E}{\partial w_{pk}^l} = \frac{\partial E}{\partial o_k^{l+1}} \frac{\partial o_k^{l+1}}{\partial s_k^{l+1}} \frac{\partial s_k^{l+1}}{\partial w_{pk}^l}$$

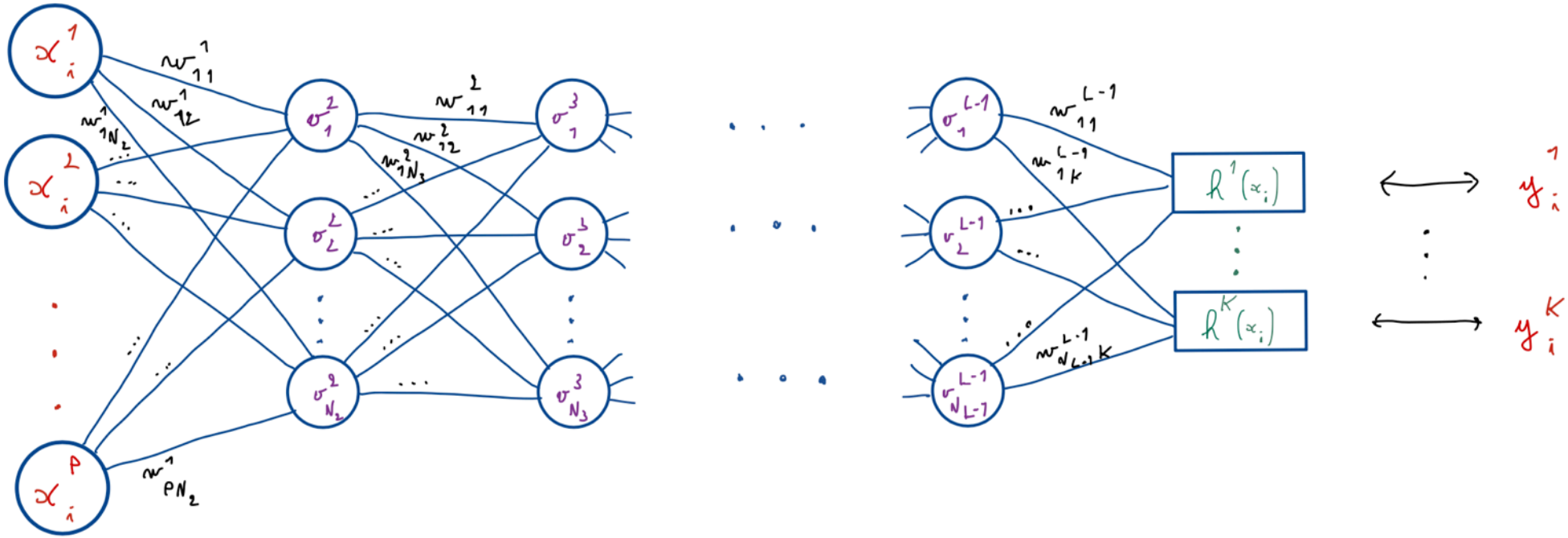
$l = 1$

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

$l = 1$

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

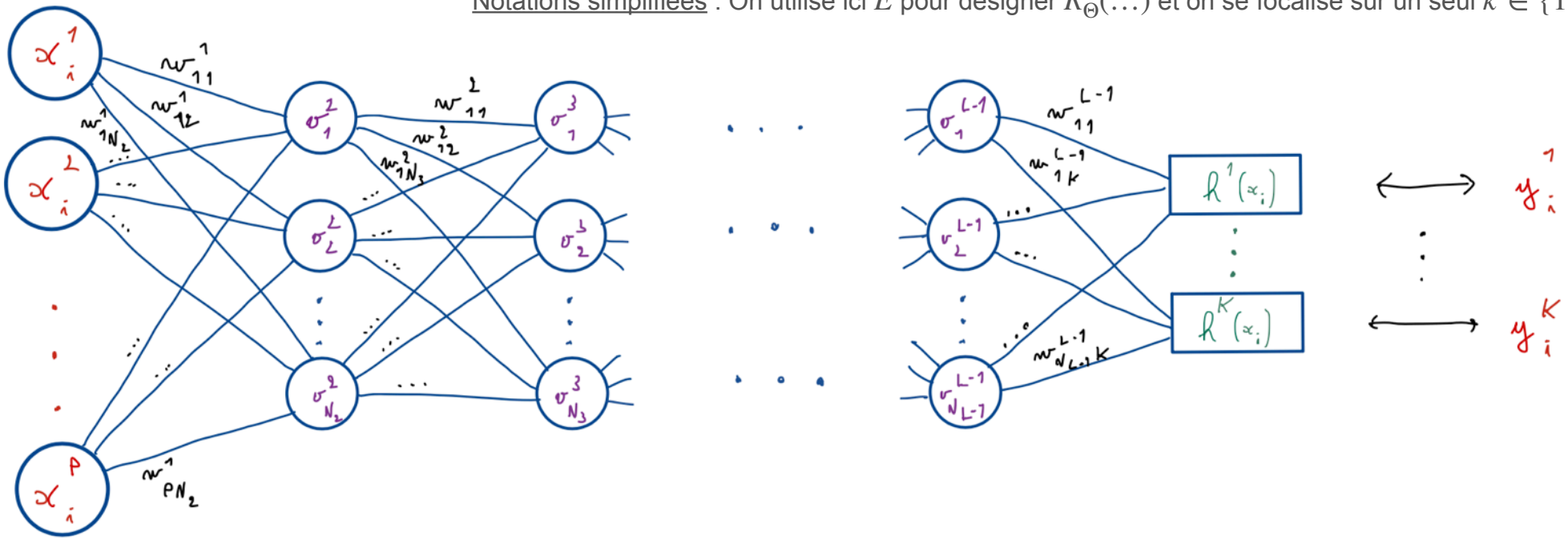
$l = 1$



BINGO !!!

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

Notations simplifiées : On utilise ici E pour désigner $R_\Theta(\dots)$ et on se focalise sur un seul $k \in \{1,\dots,K\}$

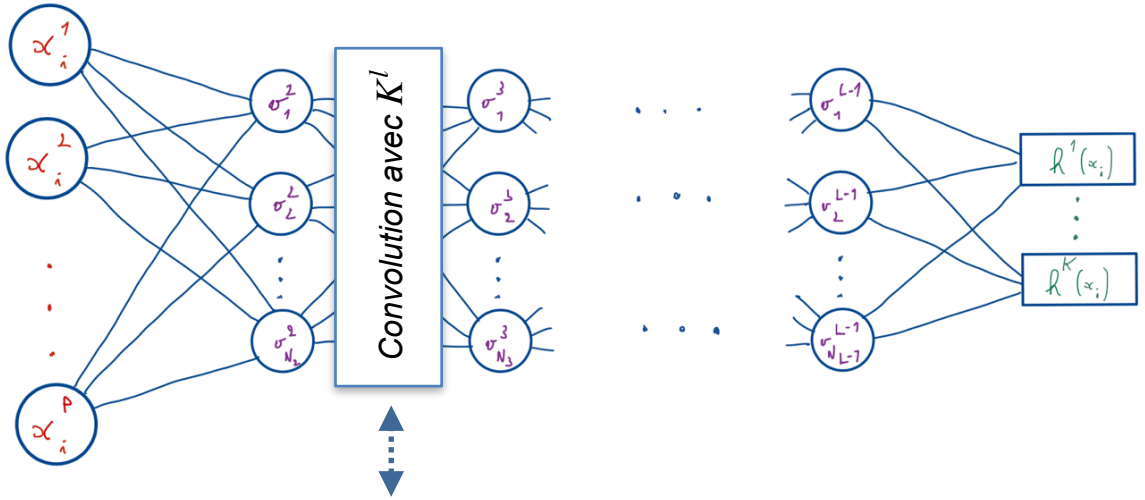


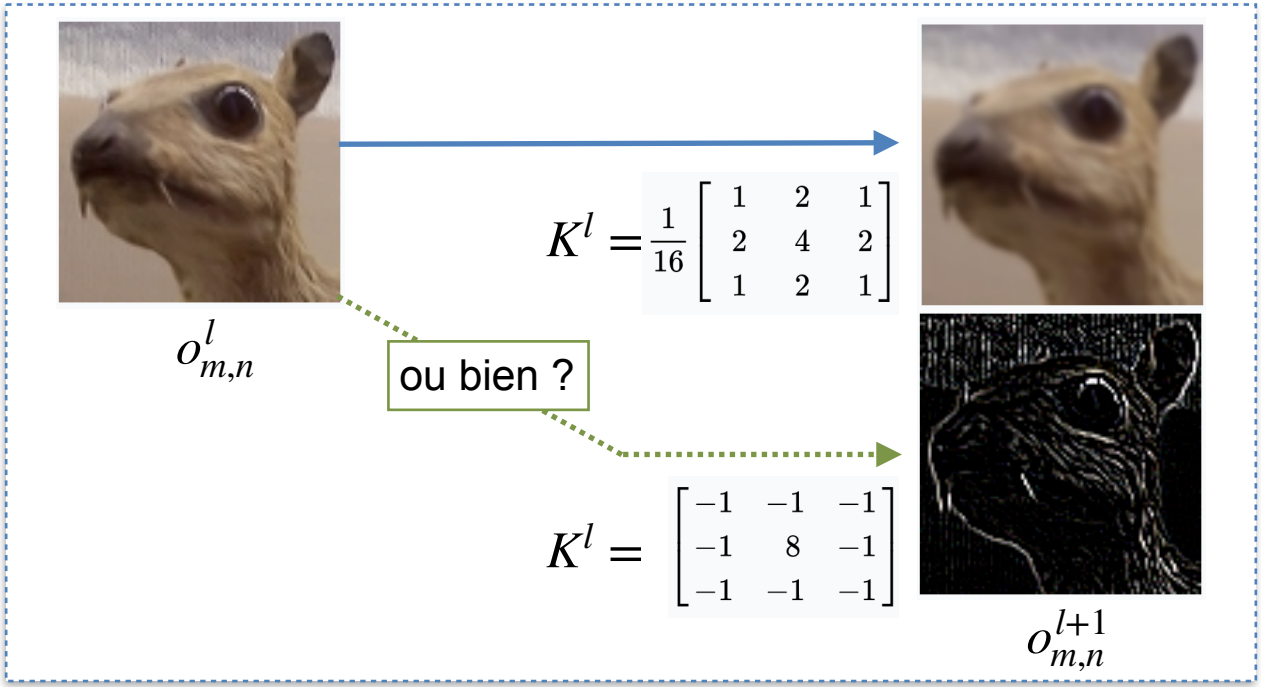
Descente de gradient pour apprendre les $\{w_{pk}^l\}_{p,k,l}$:

1: (Phase *Forward*)
2: Pour chaque observation x_i, calcule $h(x_i) = (h^1(x_i), \dots, h^K(x_i))^T$
3: Pour chaque observation x_i, calcule les $\left(\frac{\partial E}{\partial o_k^L}\right)_i$ sur la couche de sortie en utilisant les $h^k(x_i)$ et les y_i^k.
4: (Phase *Backward*)
5: Pour l valant de $L-1$ à 1 : Calcule les $\frac{\partial E}{\partial w_{pk}^l}$ et les $\frac{\partial E}{\partial o_p^l}$ en utilisant les $\frac{\partial E}{\partial o_p^{l+1}}$

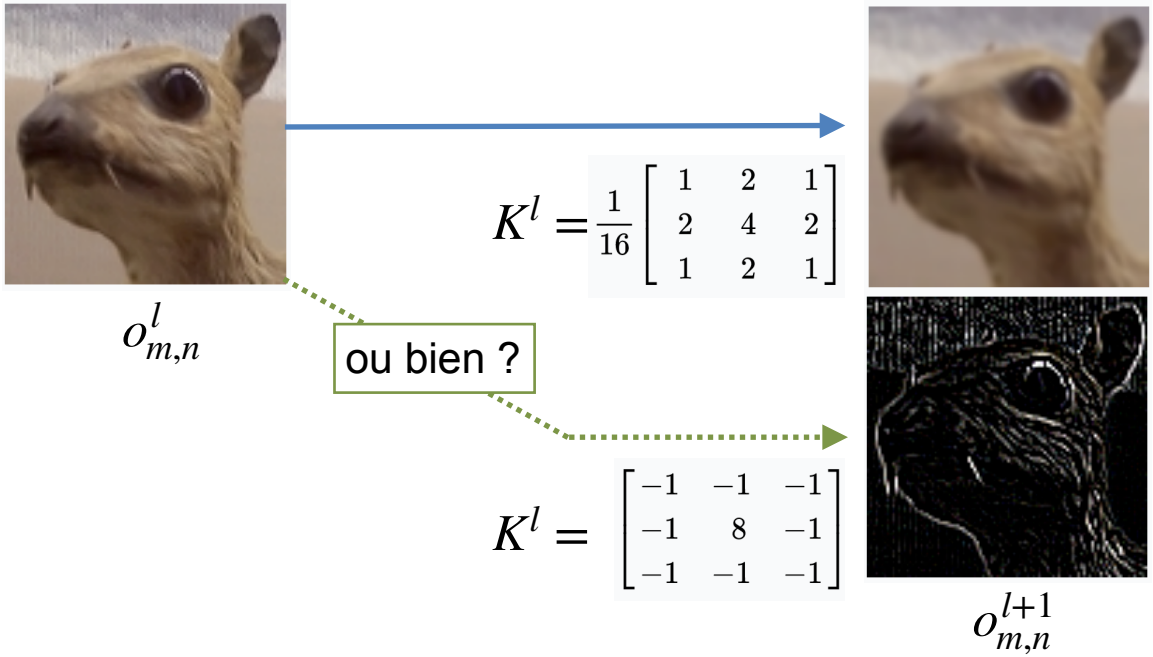
6: (Etape de descente de gradient)
7: $\forall\{p,k,l\}$: $(w_{pk}^l) = (w_{pk}^l) - \eta \frac{\partial E}{\partial w_{pk}^l}$

Couches de convolution : On peut *apprendre les poids de filtres de convolution optimaux,* plutôt que les poids de couches *denses* (ou fully connected),.





Couches de convolution : On peut *apprendre les poids de filtres de convolution optimaux,* plutôt que les poids de couches *denses* (ou fully connected),.



Modèle *forward* :

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

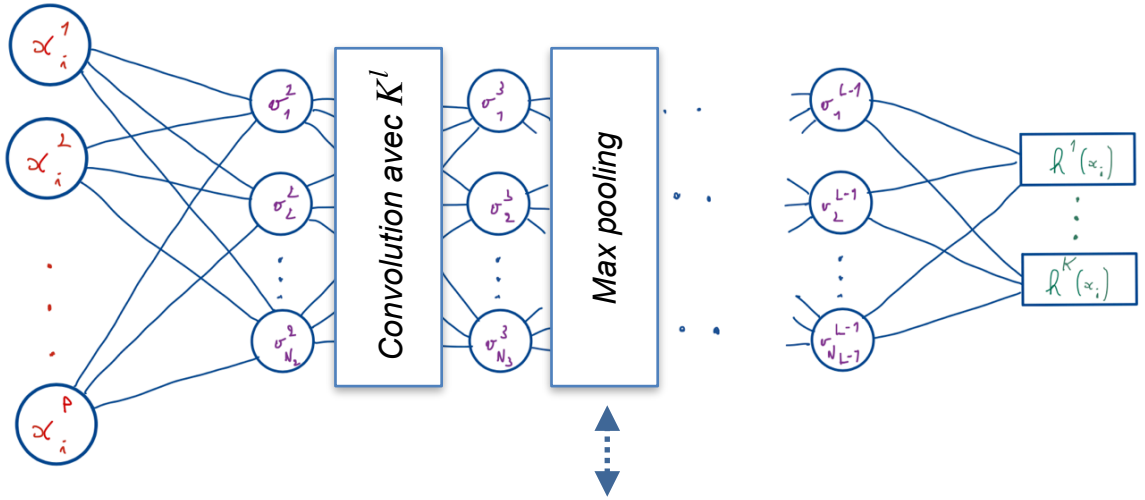
Equations *backward* :

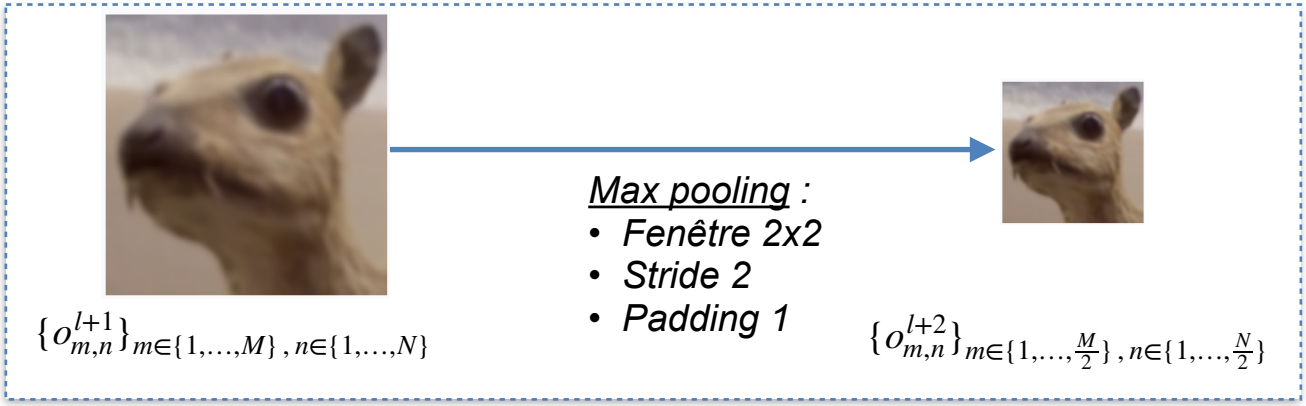
$$\frac{\partial E}{\partial o^l_{m,n}} = \sum_{v=-V}^{V} \sum_{w=-W}^{W} \frac{\partial E}{\partial o^{l+1}_{m+v,n+w}} \frac{\partial o^{l+1}_{m+v,n+w}}{\partial s^{l+1}_{m+v,n+w}} \frac{\partial s^{l+1}_{m+v,n+w}}{\partial o^l_{m,n}} = \frac{\partial E}{\partial o^{l+1}_{m+v,n+w}} \varphi'(s^{l+1}_{m+v,n+w})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

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

Remarque : Se parallélise très bien sur un GPU

Couches de max pooling : Permet de réduire la dimension des données d'une couche à la suivante





Modèle *forward* : $o^{l+1}_{f,g} = \max_{v=0,\dots,V-1\,;\,w=0,\dots,W-1} o^{l}_{Vf+v,Wg+w}$

Architecture de CNN classique :

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

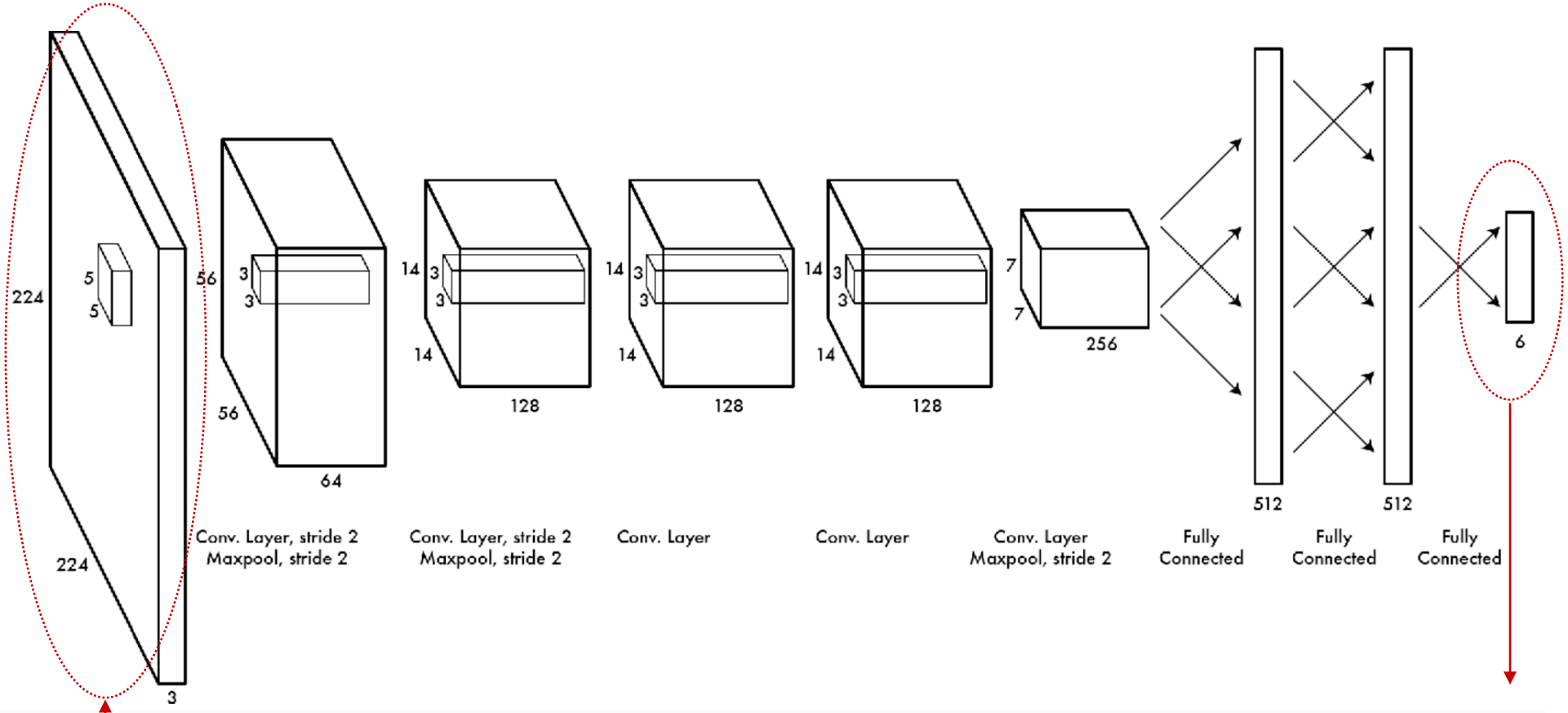


Image RGB de taille 224x224.

Exemple $x_i =$ 

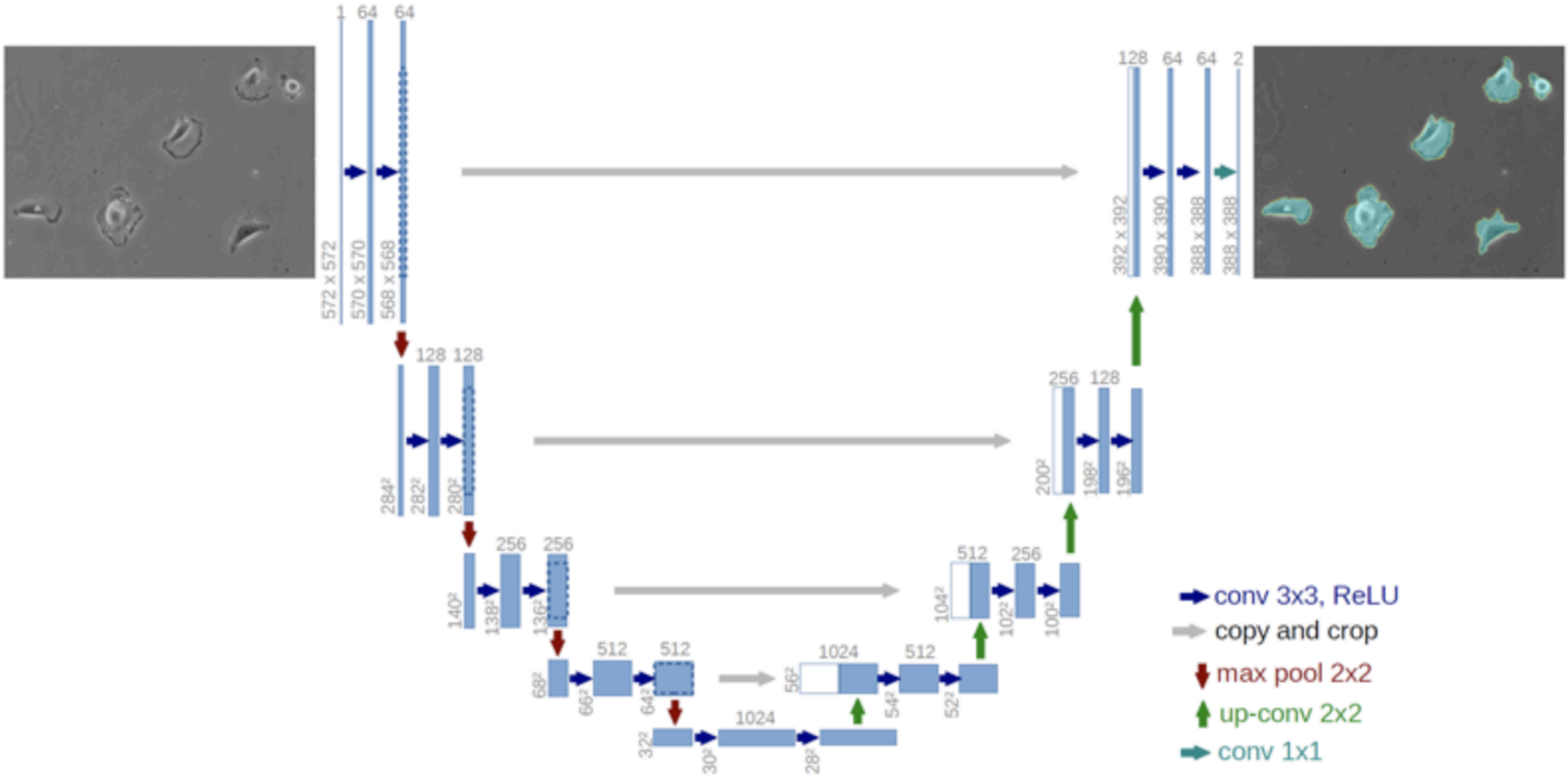
Caractéristiques de l'image.

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

Lunettes … Chapeau

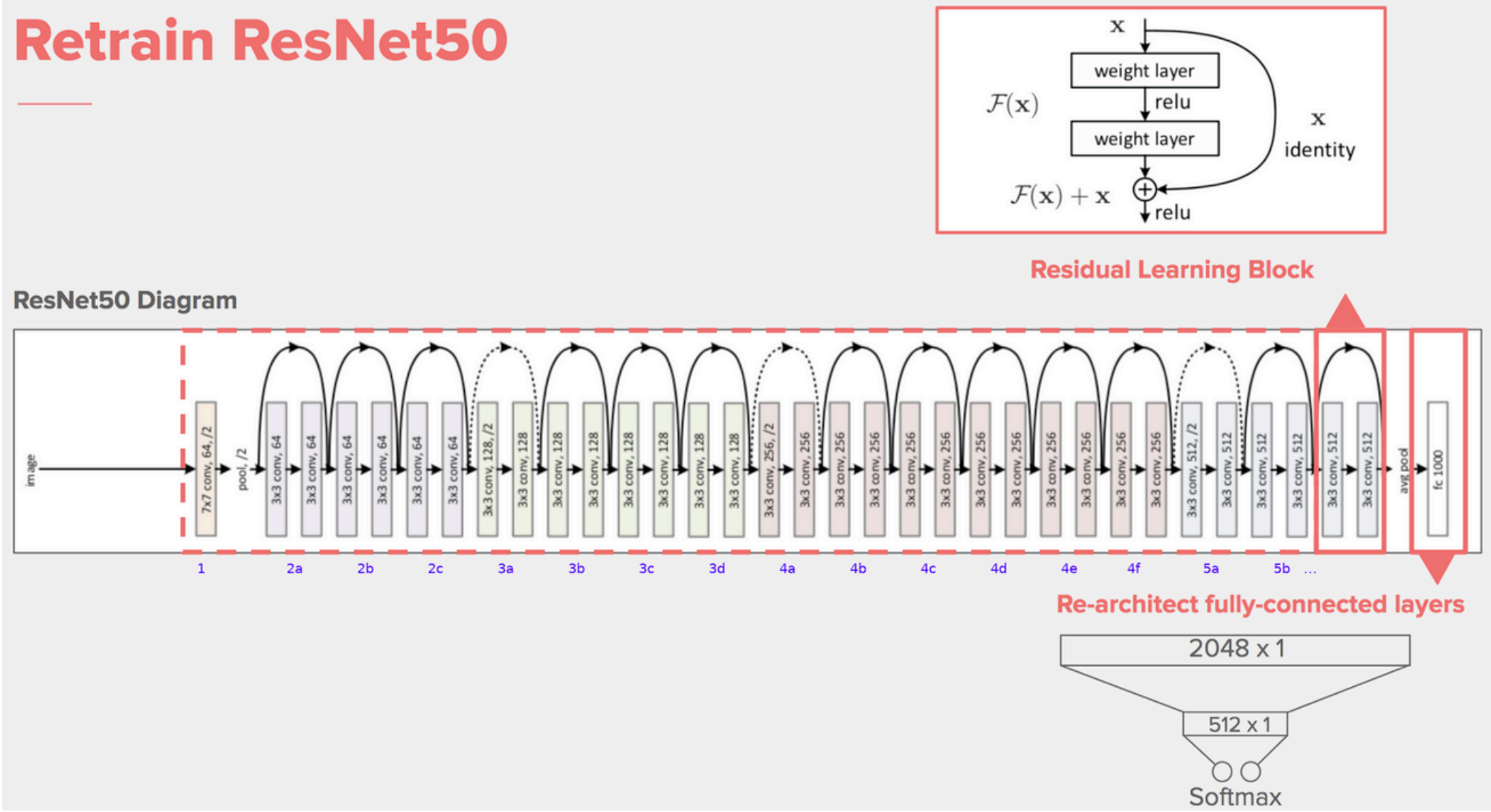
Architecture U-NET :

Image du réseau : [Ronneberger et al., MICCAI 2015]



Architecture ResNet :

Image du réseau : ResNet50



MERCI !

Passons maintenant à