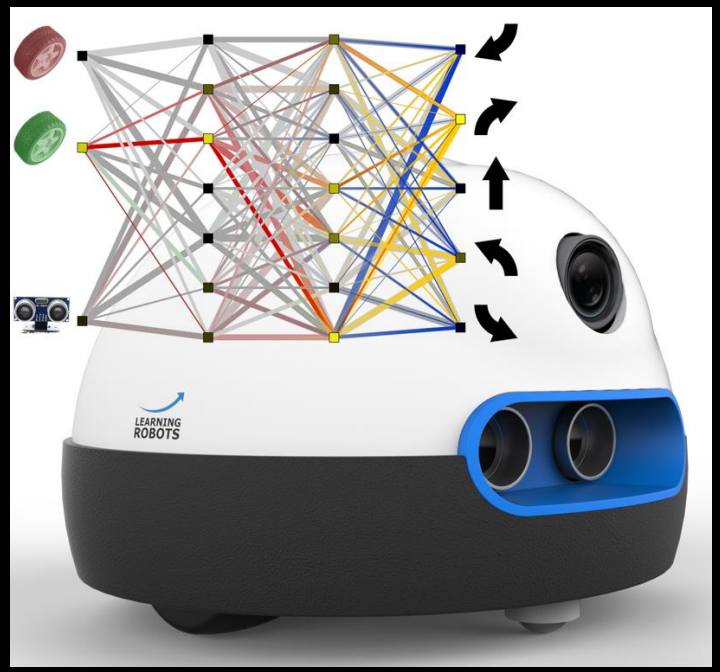
Learning Robots

Artificial Intelligence within everyone's reach

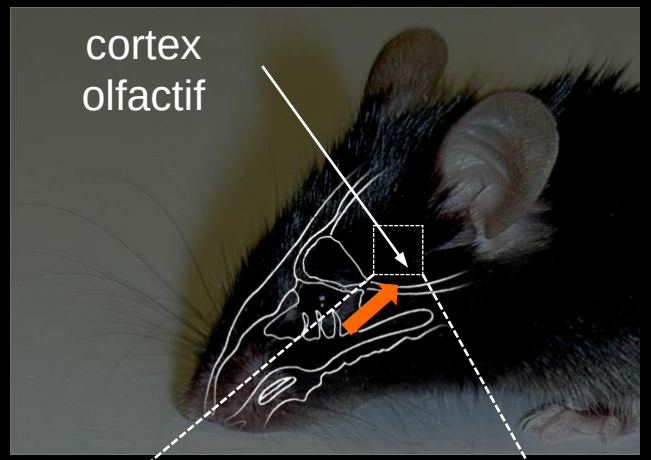
Réseaux de Neurones Artificiels

(Artificial Neural Network, ANN)

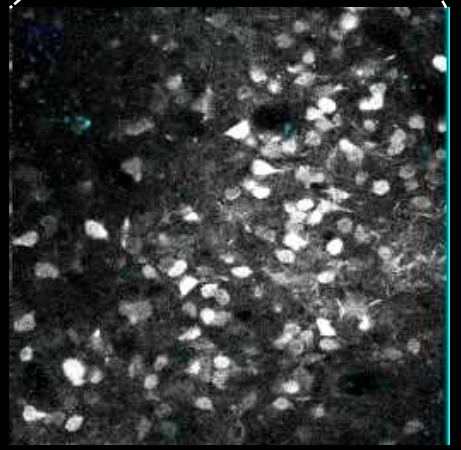
Un mot sur les neurones biologiques… et artificiels



Les neurones répondent aux stimuli



Odeur

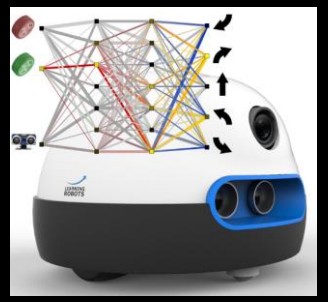


Stimulus

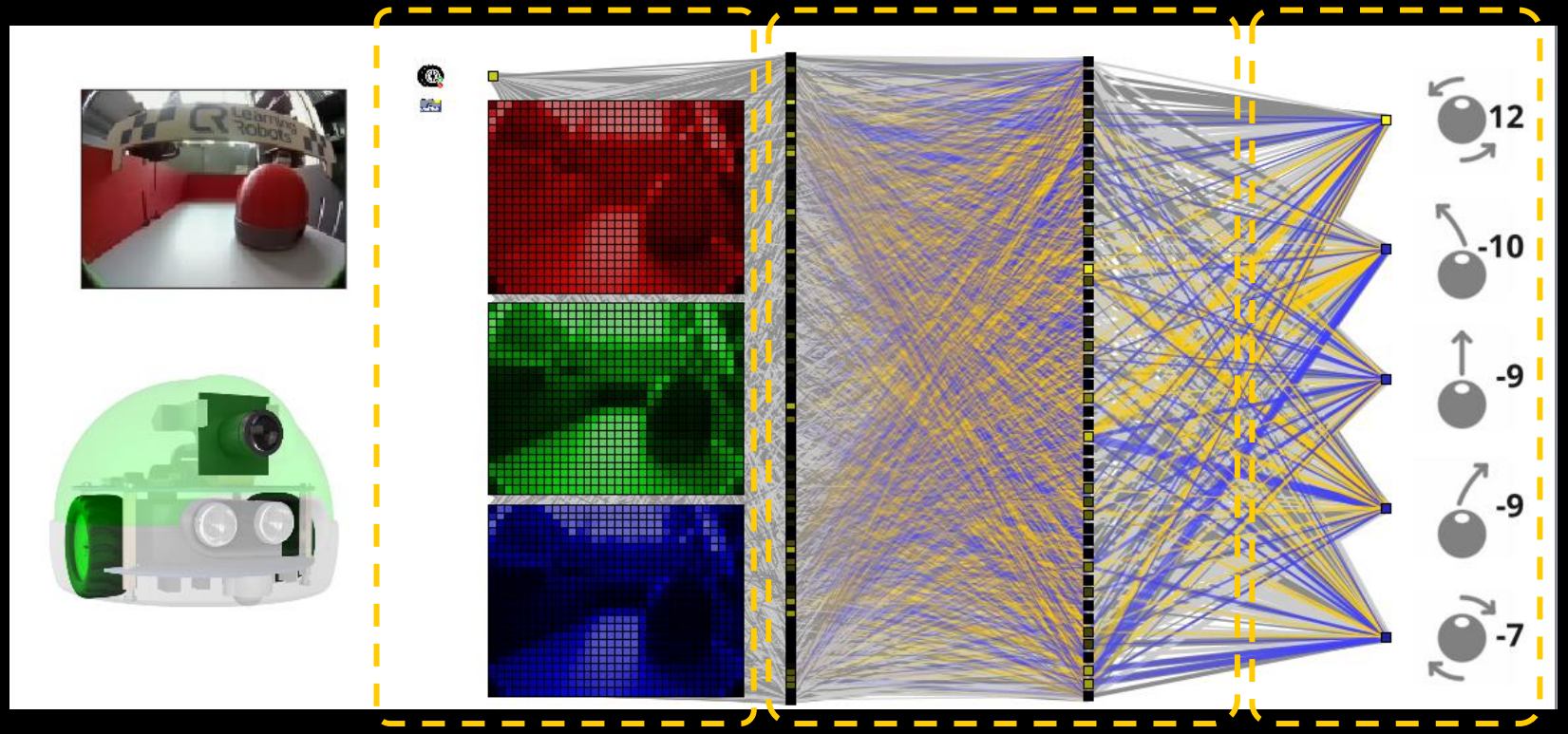
Neurones

Action

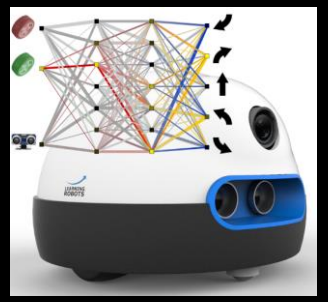
Réseau de neurones artificiels (“Deep Learning”)



Capteurs → « Cerveau » → Action

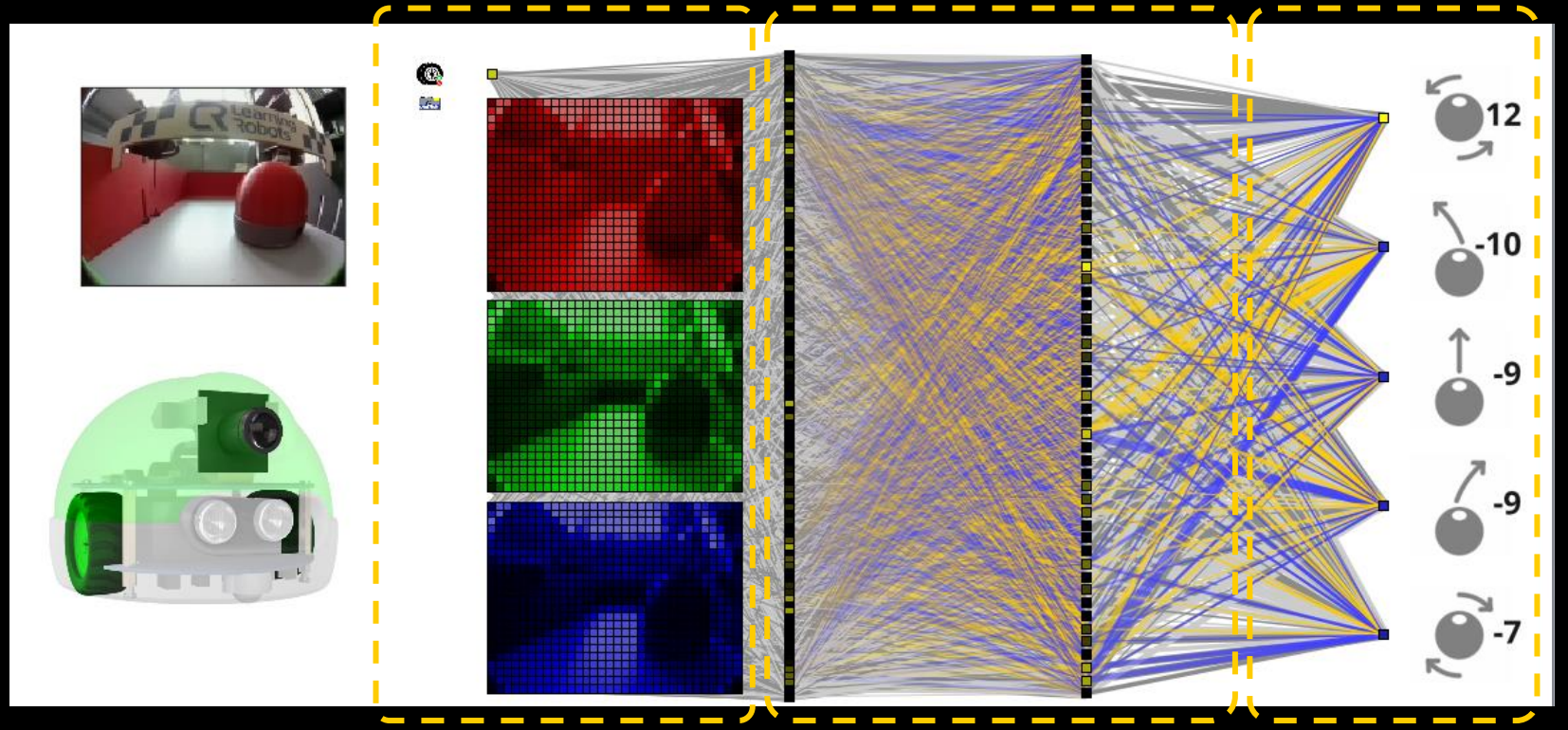


Apprentissage supervisé du réseau de neurones

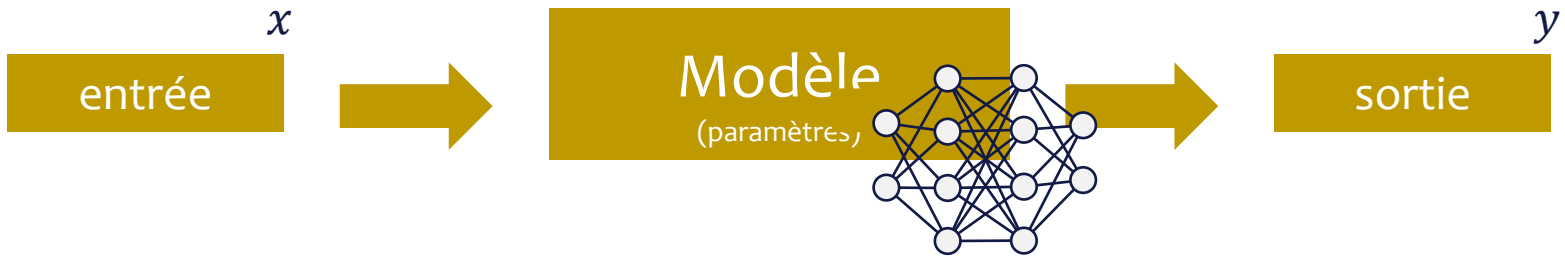


Capteurs → « Cerveau » → Action

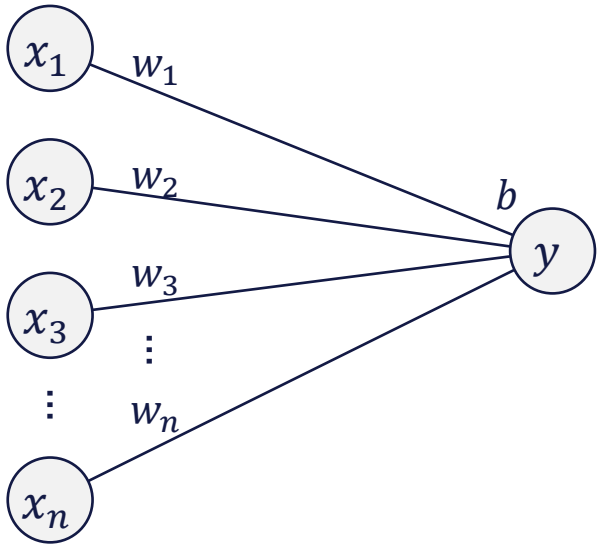
Apprentissage



Le *Modèle* en Machine Learning : détails pour le réseau de neurones

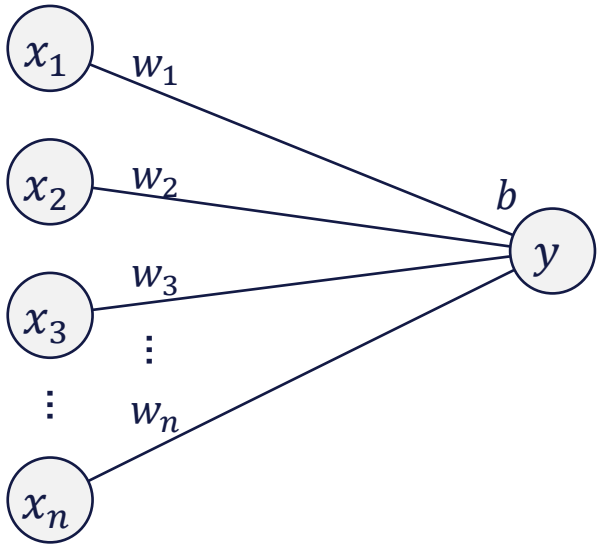


Etude de 1 neurone (= le perceptron)



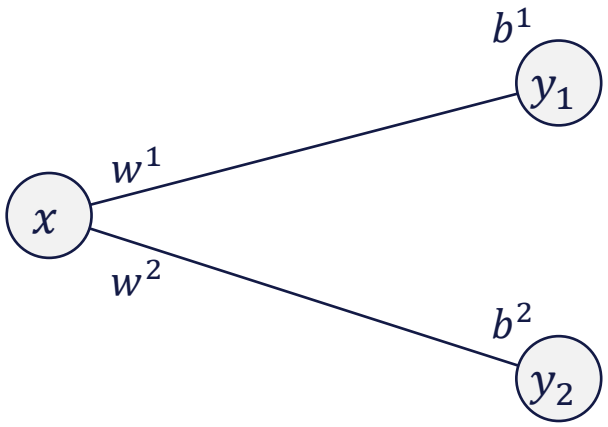
$$y = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

Etude de 1 neurone (= le perceptron)



$$y = \sum_i w_i x_i + b$$

Illustration avec 1entrée, 2 sorties



$$y_1 = w^1 x + b^1$$

$$y_2 = w^2 x + b^2$$

$$\text{decision} = \underset{j}{\text{argmax}}\, y_j$$

La relation entrée / sortie est une équation de droite

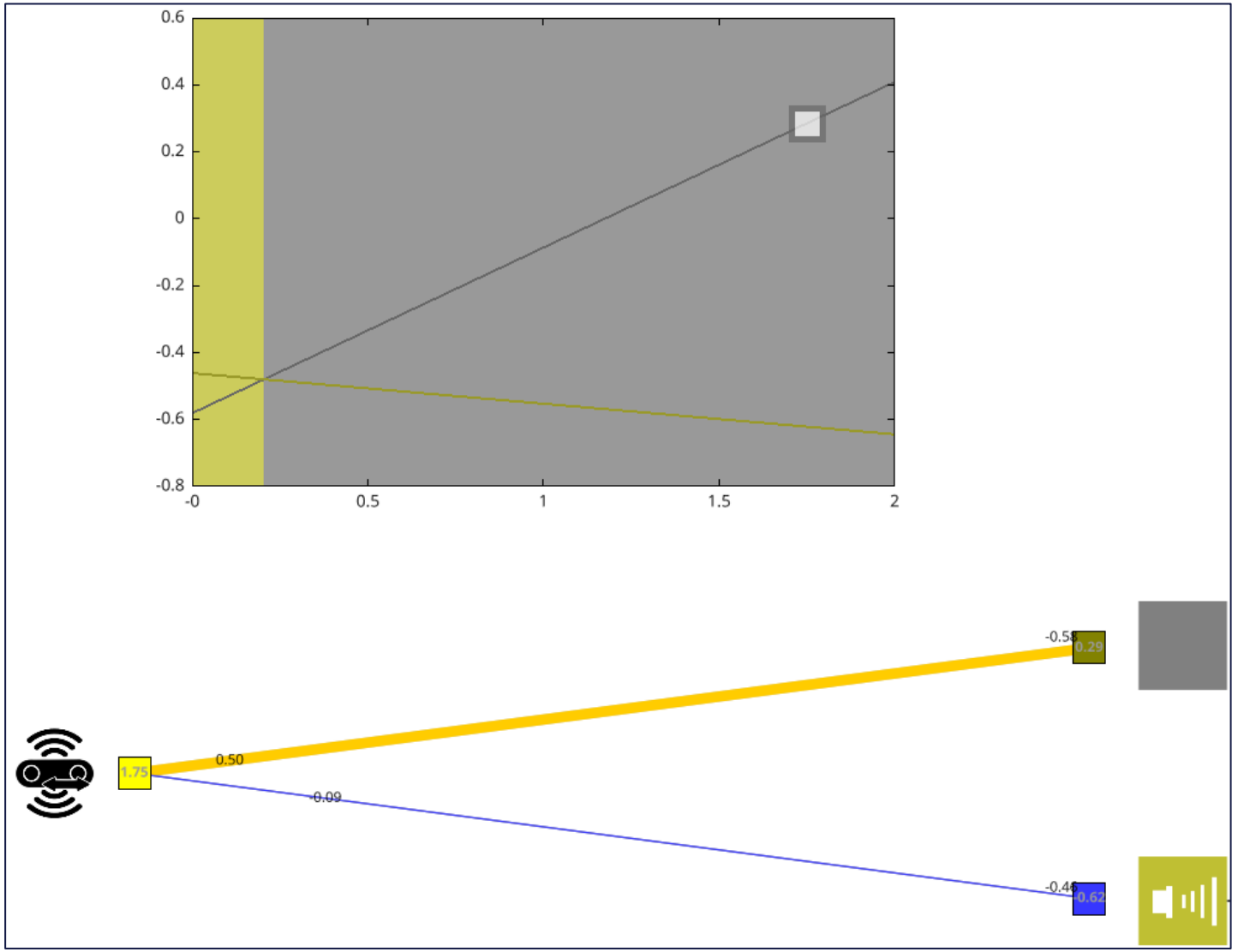
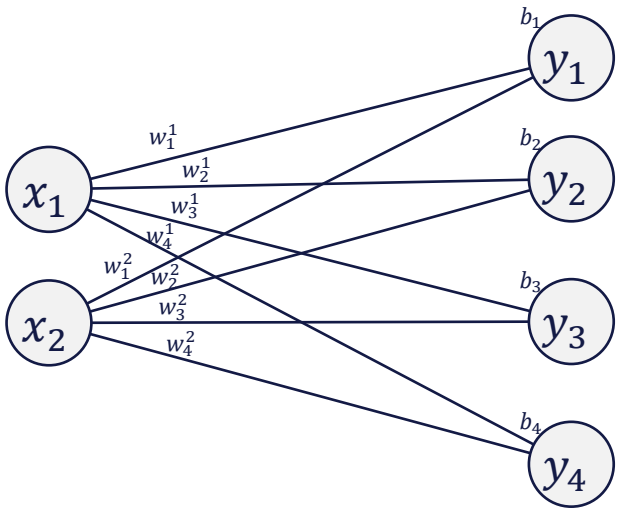


Illustration avec 2 entrées, 4 sorties



$$y_j = w_1^j x_1 + w_2^j x_2 + b_j$$

$$\text{decision} = \underset{j}{\operatorname{argmax}}\, y_j$$

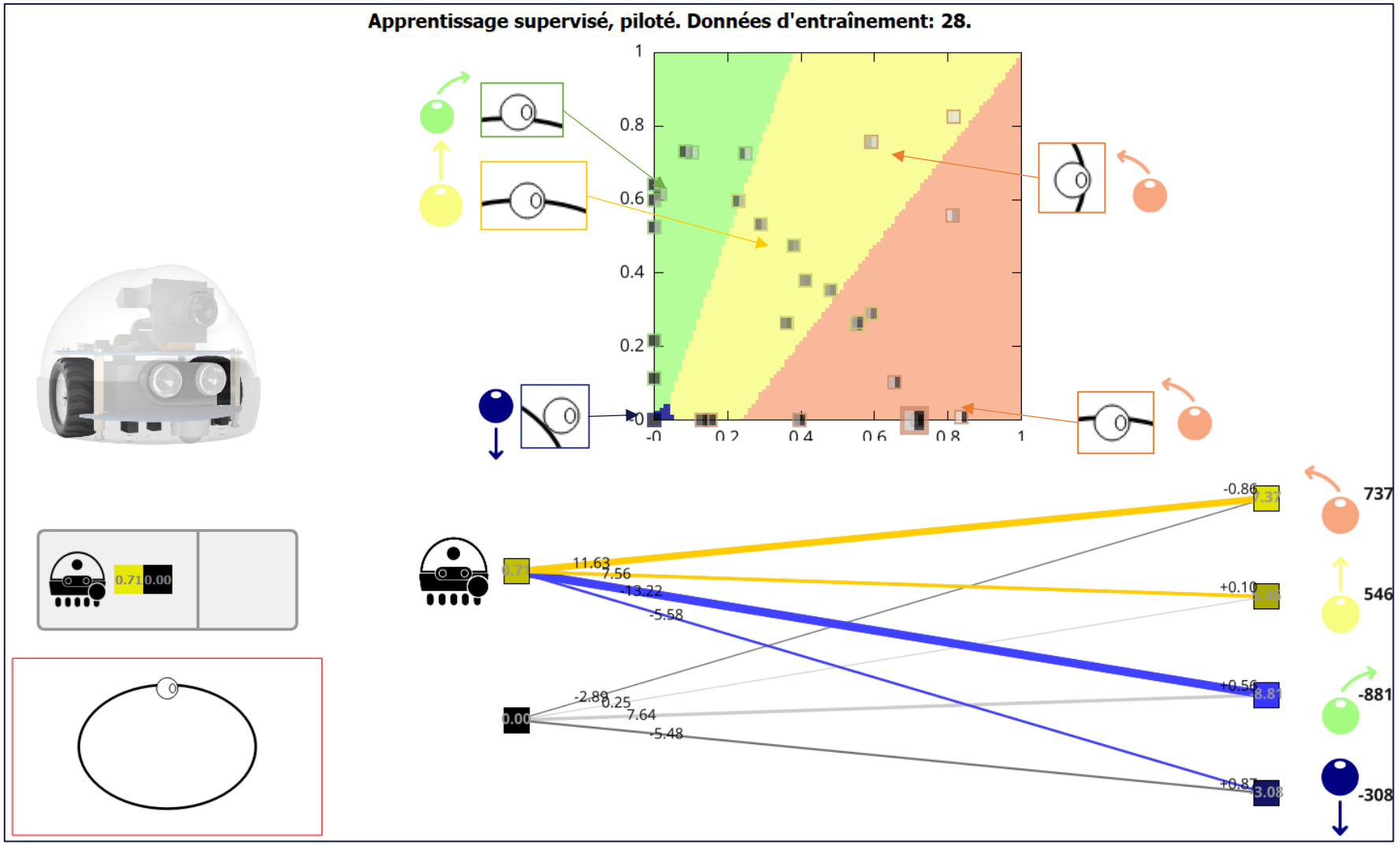
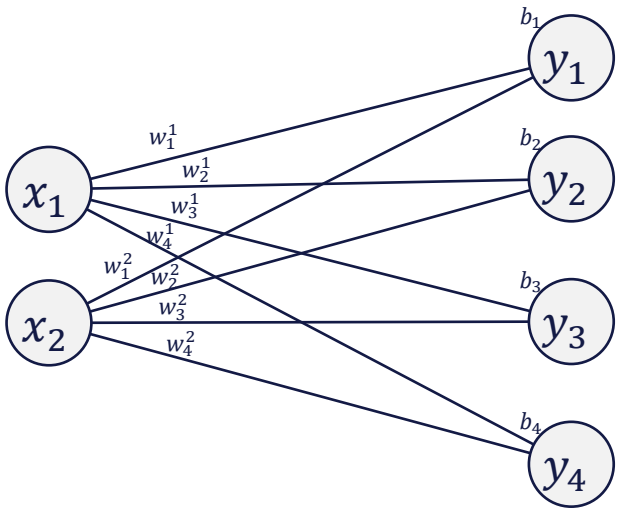
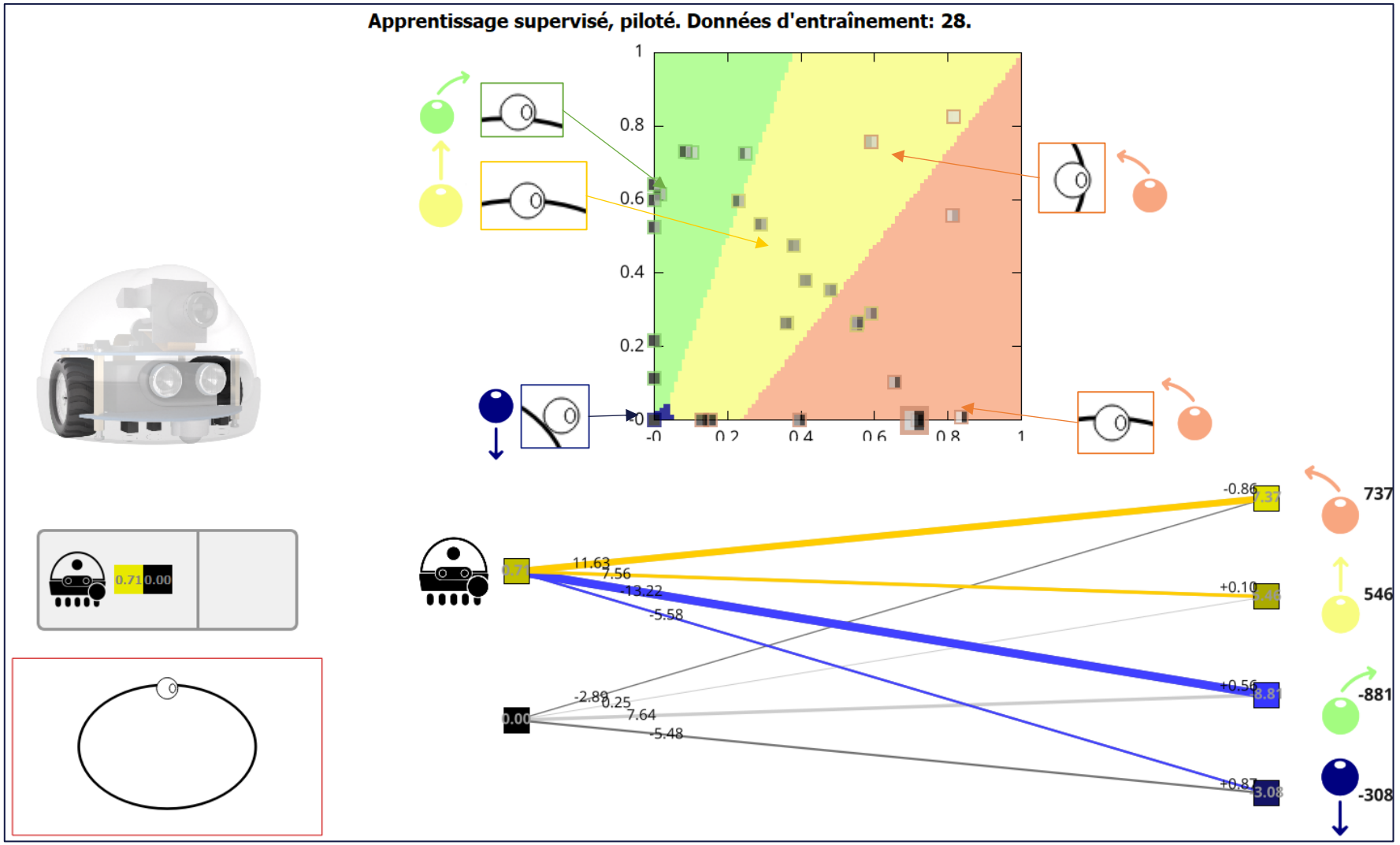


Illustration avec 2 entrées, 4 sorties

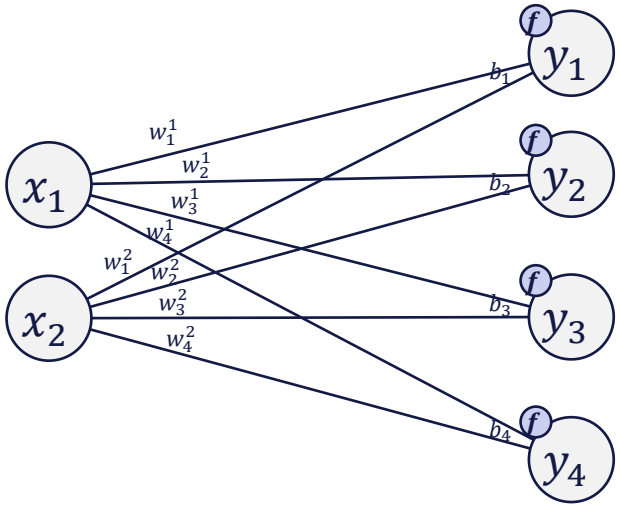


$$y_j = \sum_i w_i^j x_i + b_j$$

$$\text{decision} = \underset{j}{\operatorname{argmax}}\, y_j$$



Introduction d'une non-linéarité : la fonction d'activation



$$y_j = f\left(\sum_i w_i^j x_i + b_j\right)$$

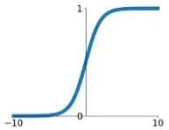
$$\text{decision} = \underset{j}{\text{argmax}}\, y_j$$

f = activation function,

for example :

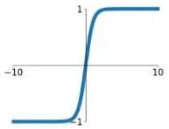
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



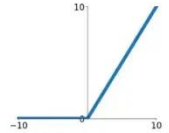
tanh

$\tanh(x)$



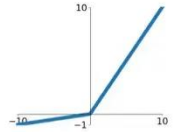
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

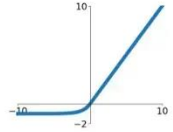


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

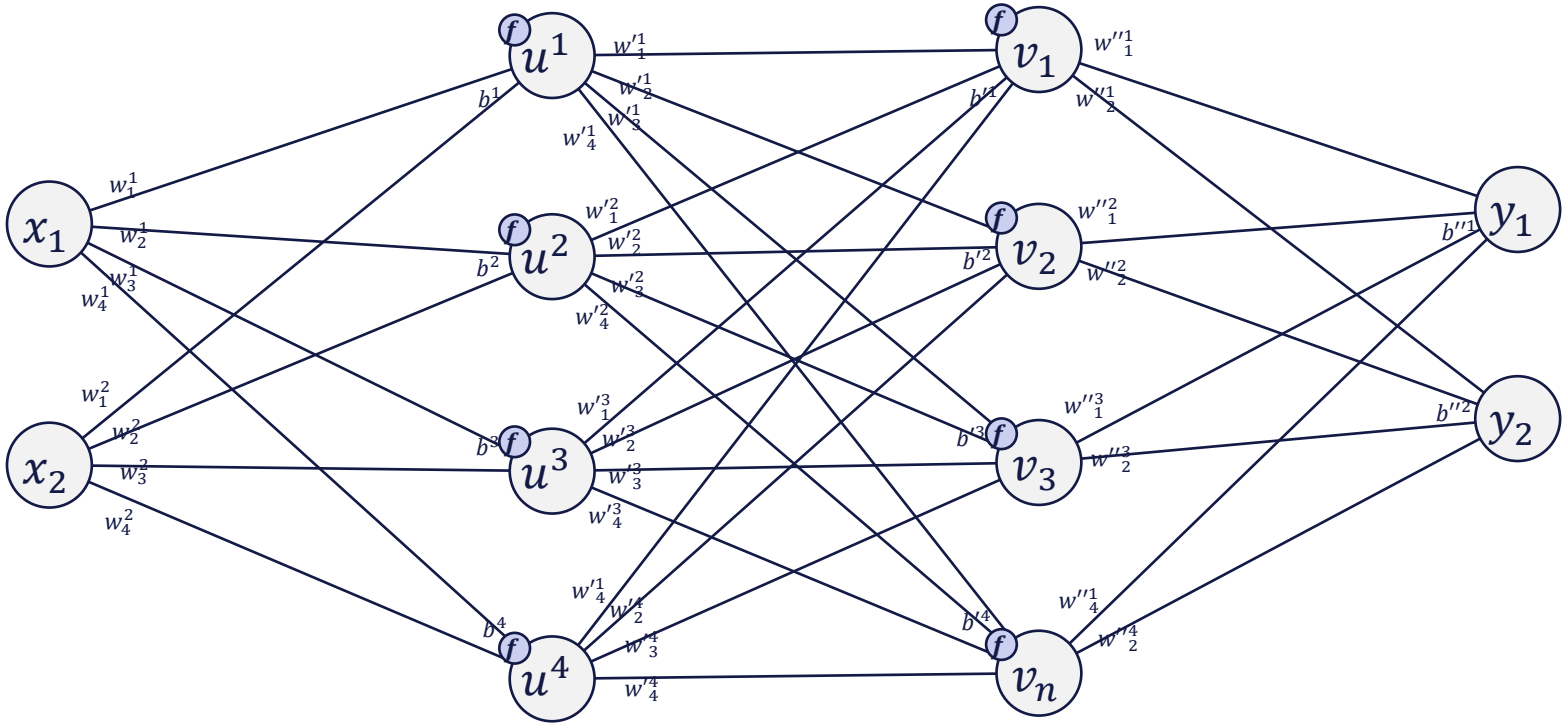
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



https://medium.com/@shrutijadon/survey-on-activation-functions-for-deep-learning-9689331ba092

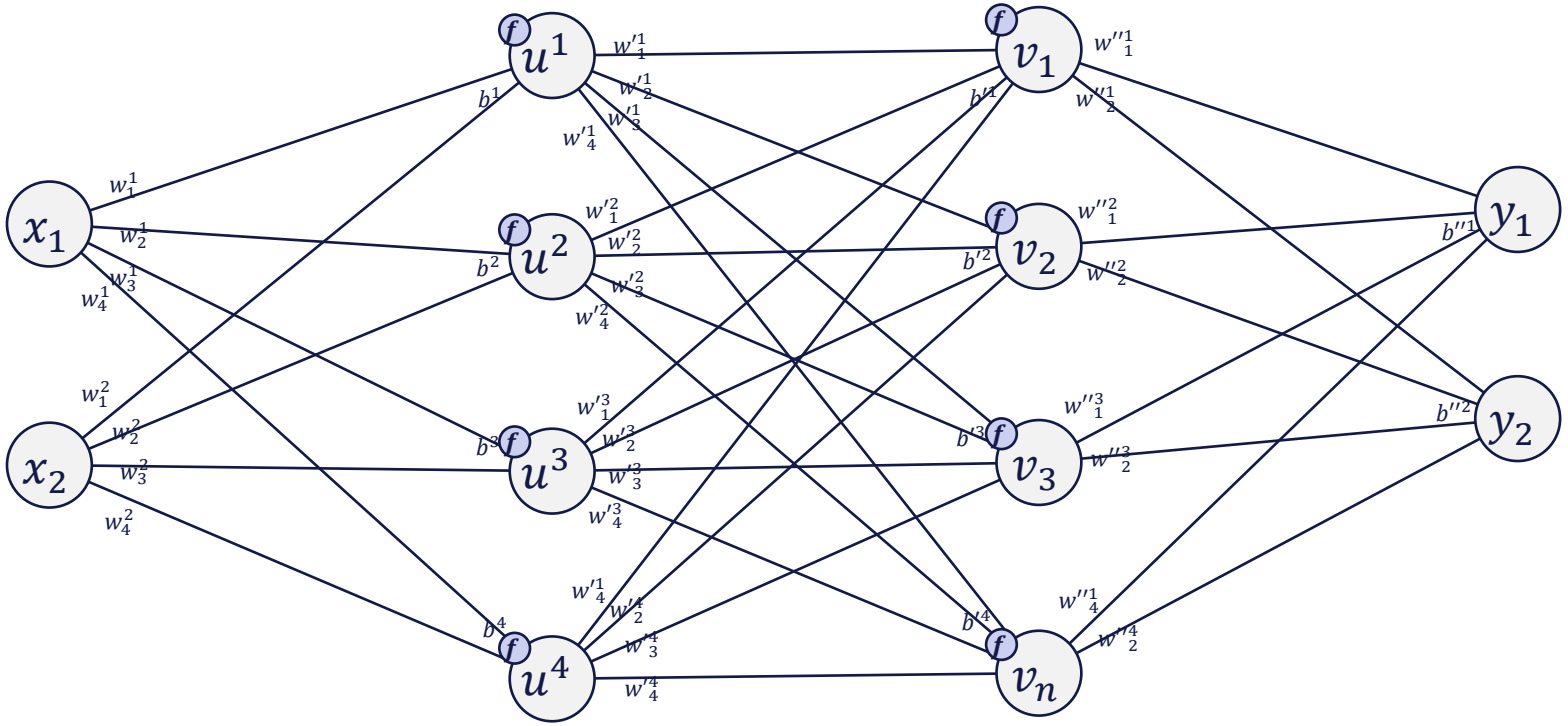
Réseau multi-couche



$$u_j = f\left(w_1^j x_1 + w_2^j x_2 + \cdots + b_j\right) \qquad v_k = f\left(w'^k_1 u_1 + w'^k_2 u_2 + \cdots + b'_j\right) \qquad y_l = w''^l_1 v_1 + w''^l_2 v_2 + \cdots + b''_l$$

$$\text{decision} = \underset{l}{\operatorname{argmax}}\, y_l$$

Réseau multi-couche



$$u_j = f\left(\sum_i w_i^j x_i + b_j\right)$$

$$v_k = f\left(\sum_j w'^k_j u_j + b'_j\right)$$

$$y_l = \sum_k w''^l_k x_k + b''_l$$

$$\text{decision} = \underset{l}{\operatorname{argmax}}\, y_l$$

Illustration avec 1entrée, 2 sorties

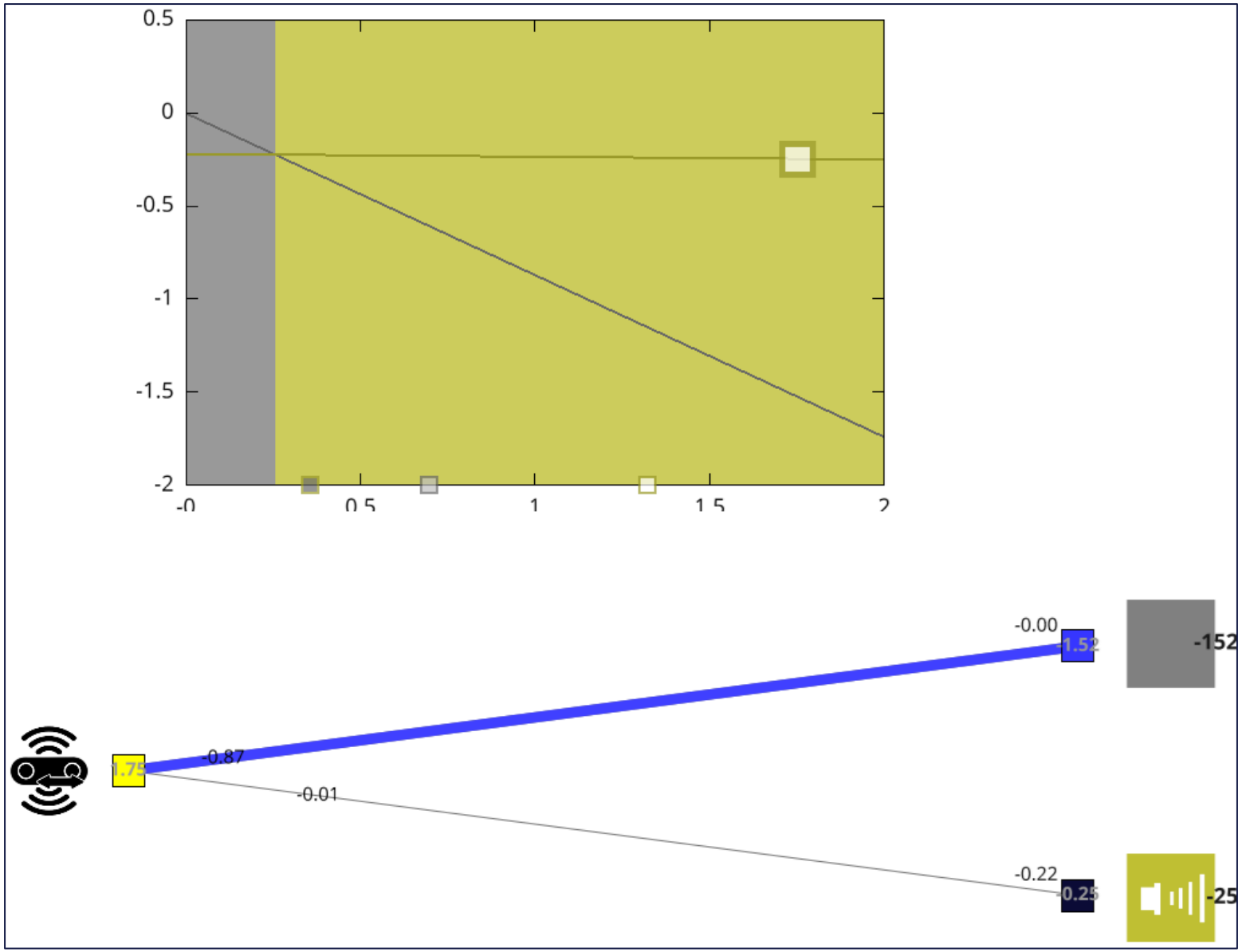


Illustration avec 1entrée, 2 sorties

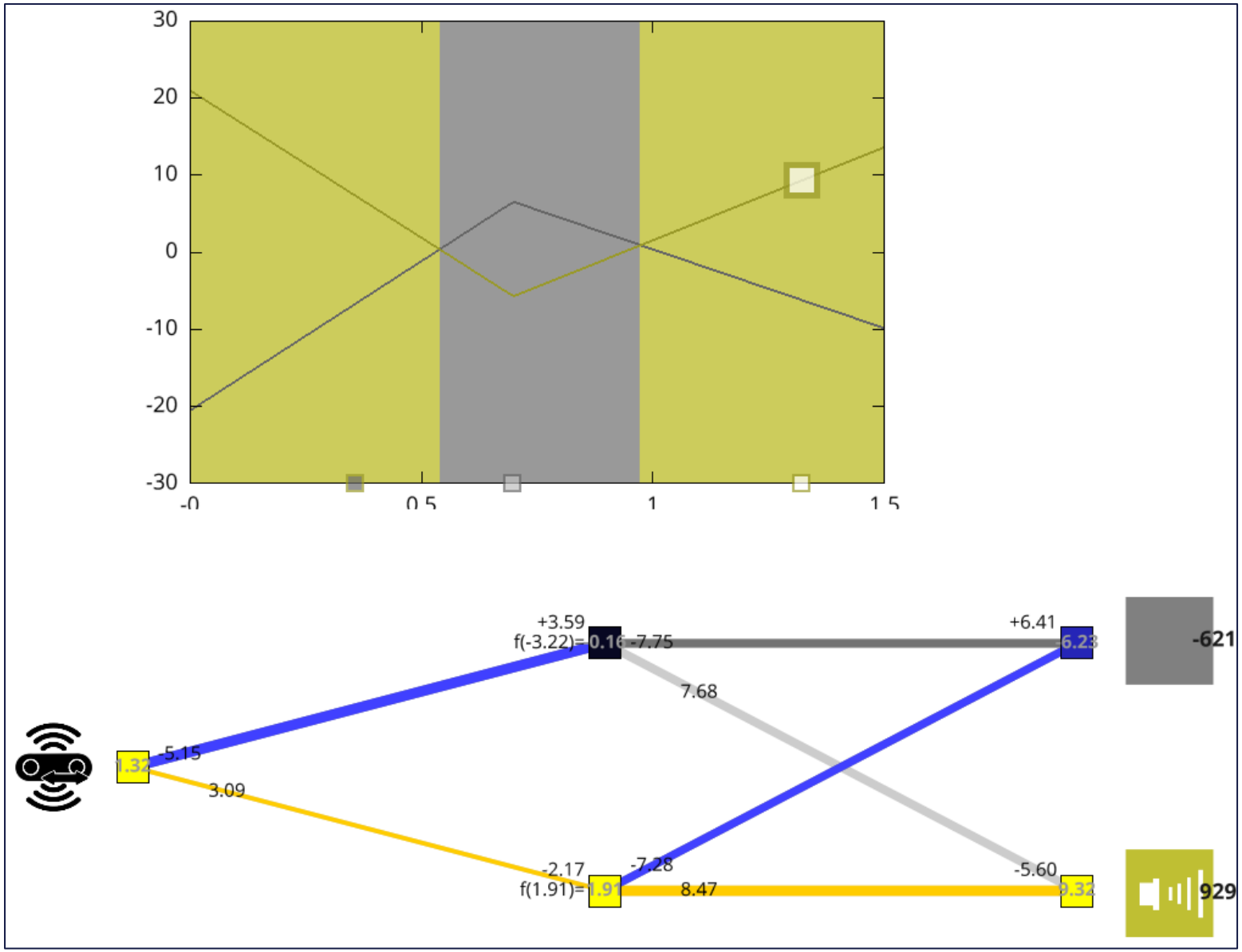


Illustration avec 2 entrées, 4 sorties

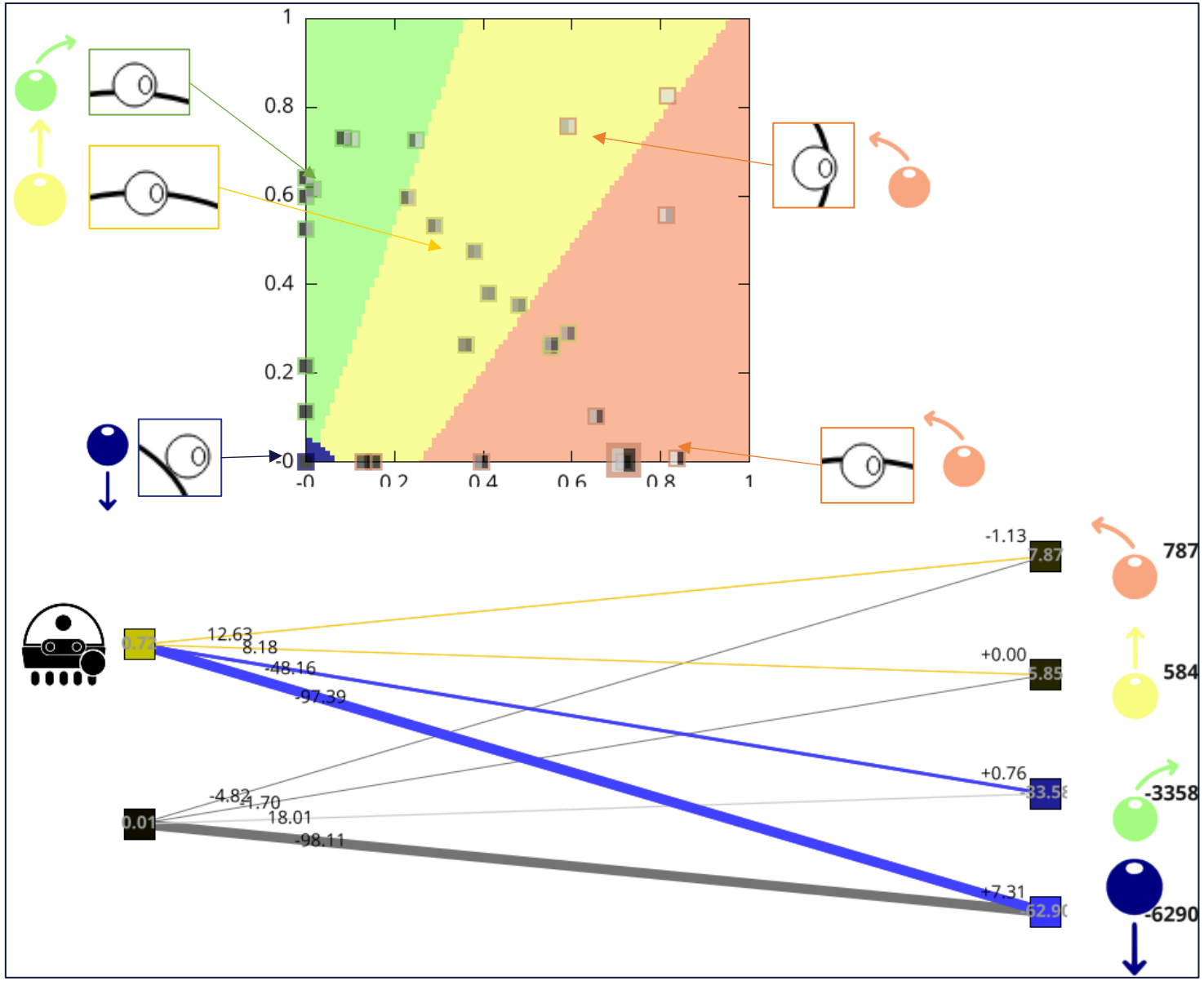


Illustration avec 2 entrées, 4 sorties

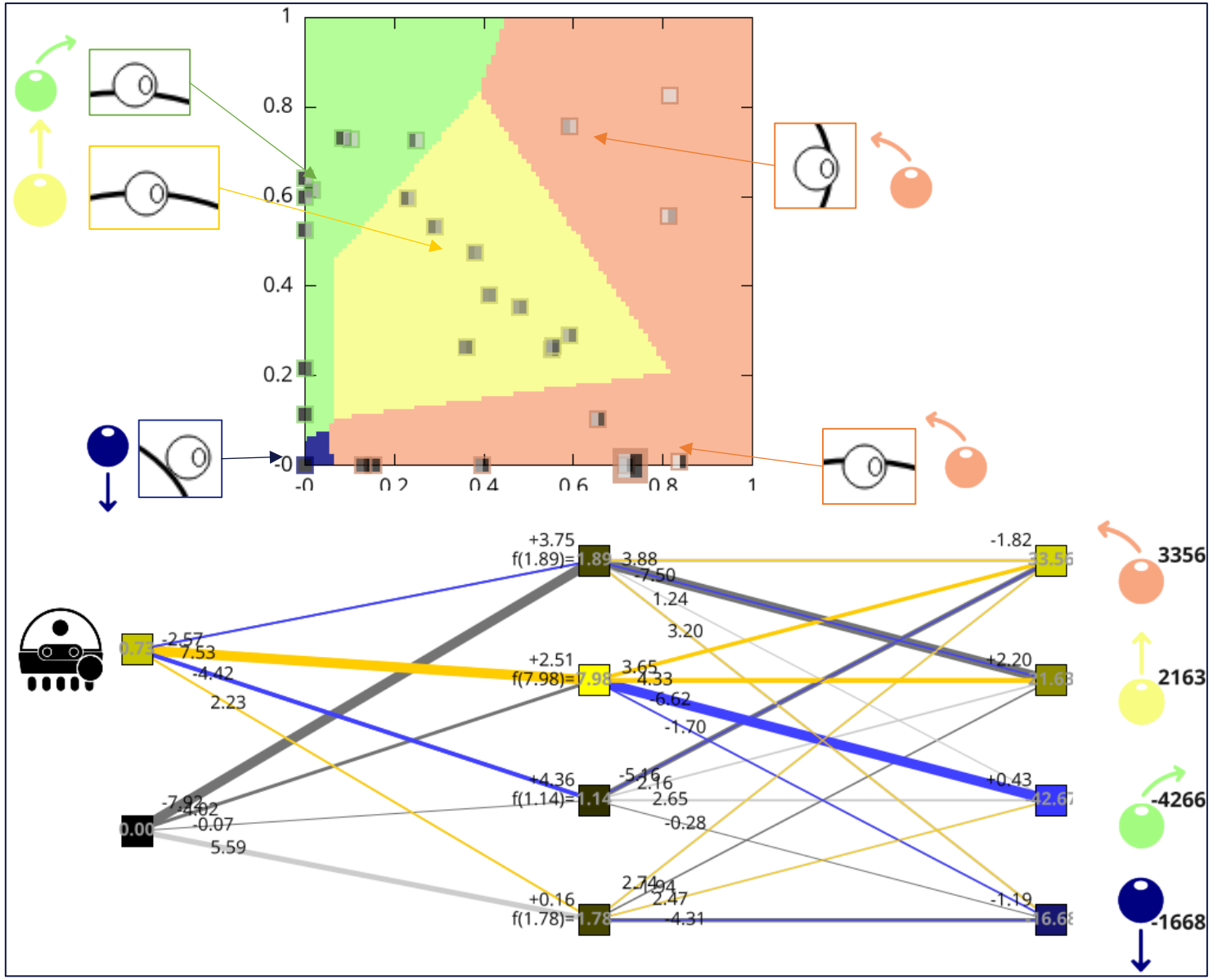
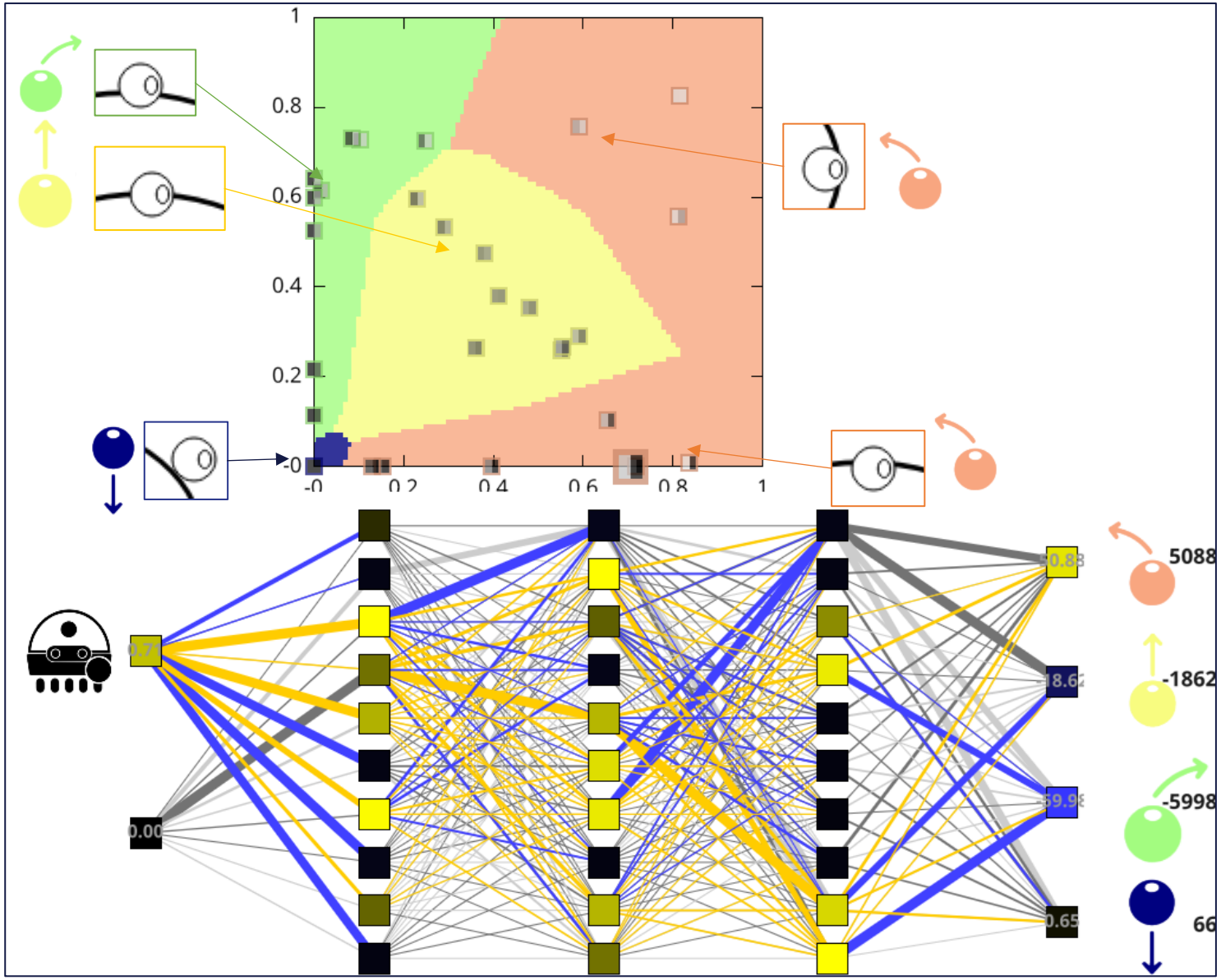
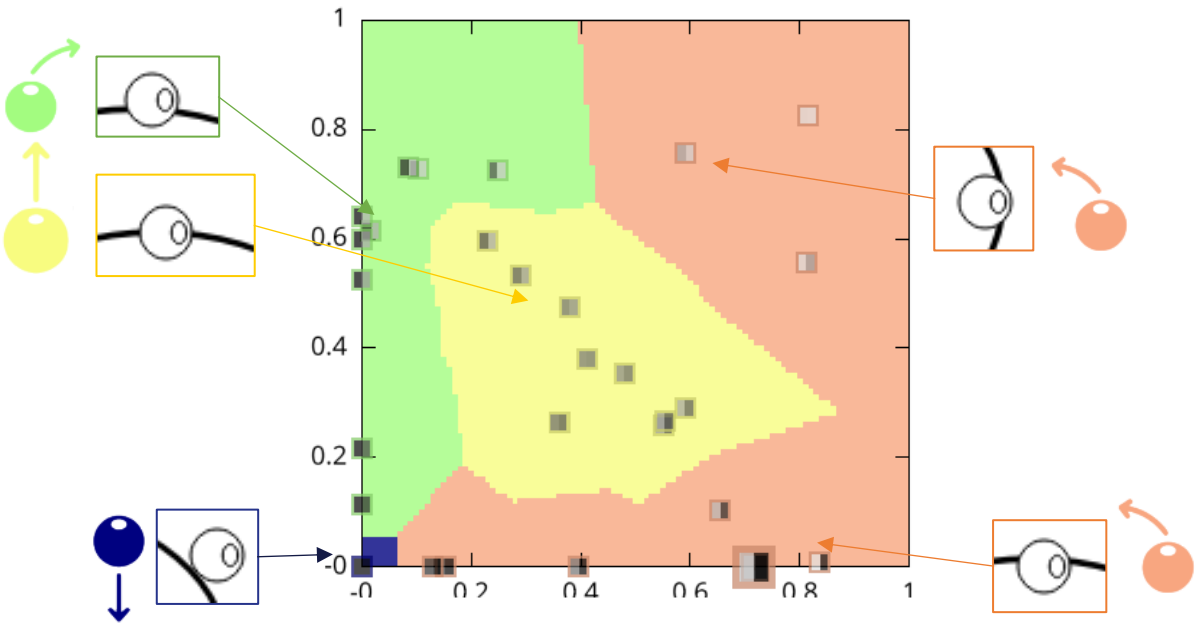


Illustration avec 2 entrées, 4 sorties

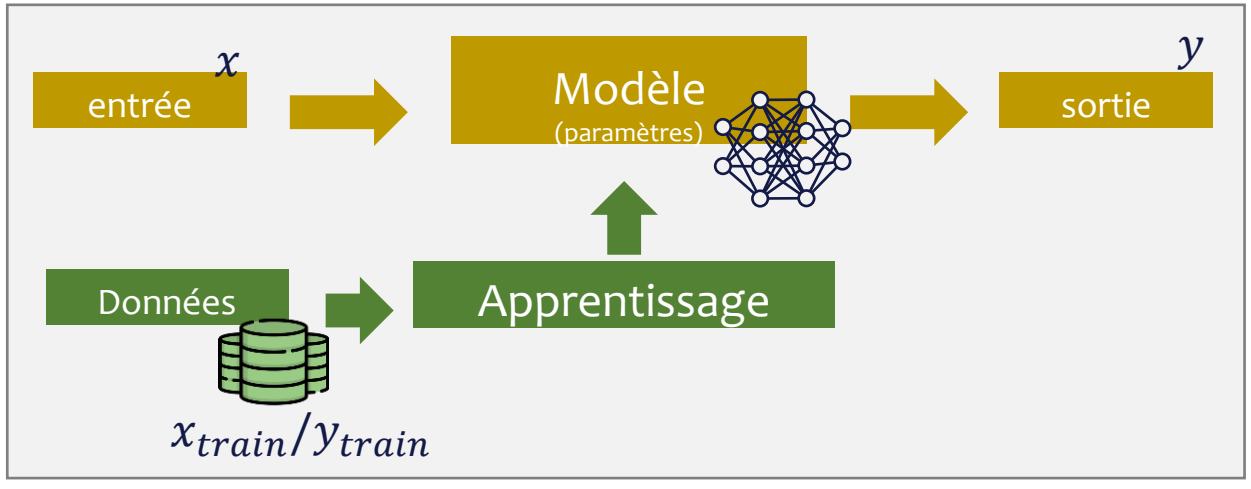


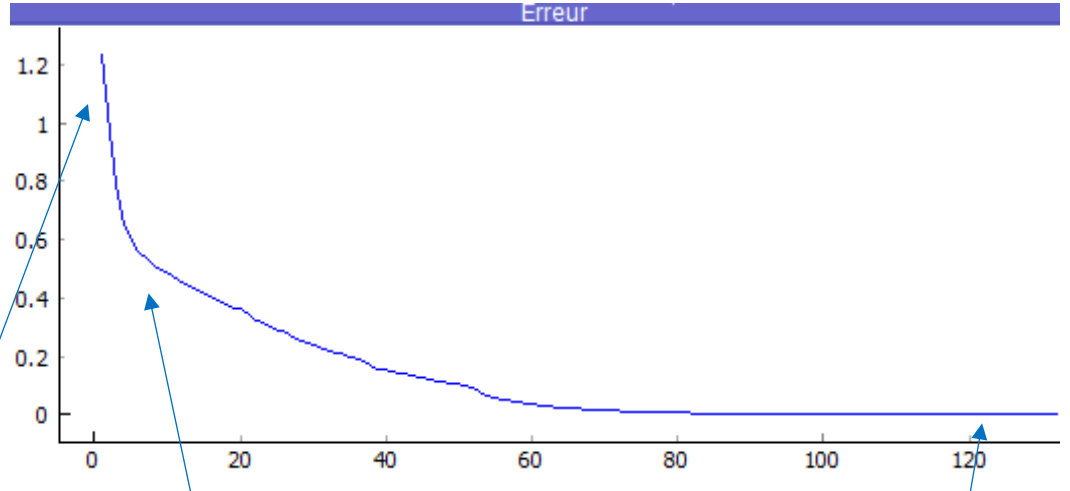
Pour comparaison, KNN



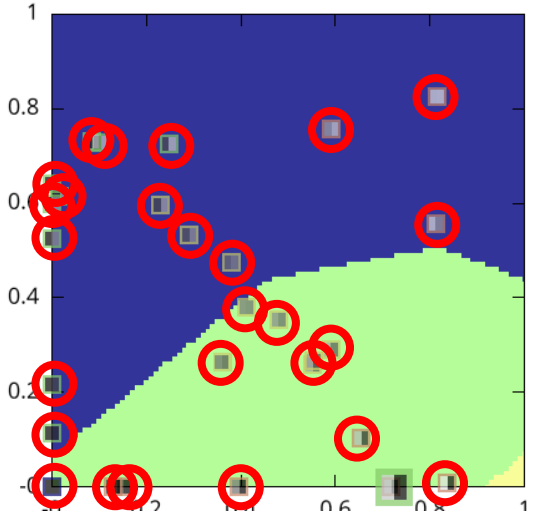
L’apprentissage du réseau de neurone

L’apprentissage consiste à modifier le modèle de manière à minimiser *l’erreur* du modèle sur les données d’entraînement.

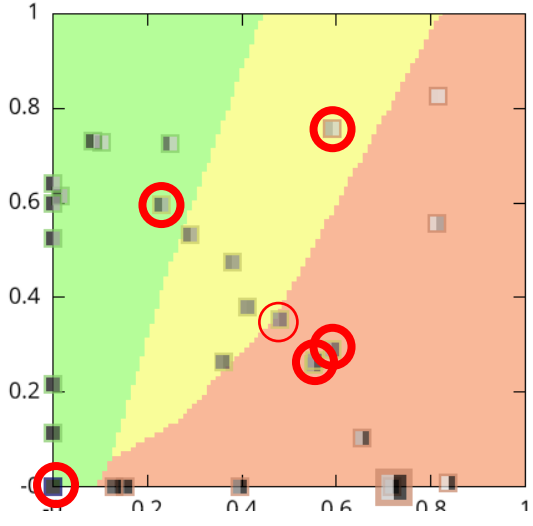




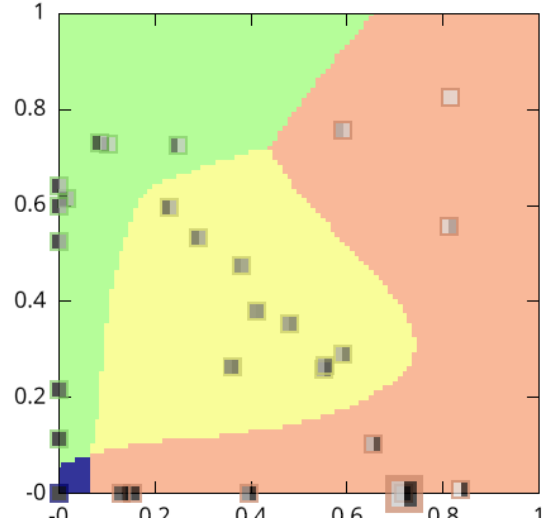
Initialement, erreur très importante



Puis l’erreur diminue



En fin d’apprentissage, l’erreur est faible



Et l'apprentissage de KNN ??

Dans l'algorithme KNN, le Modèle s'appuie toujours sur les données. L'apprentissage consiste juste à mémoriser ces données !!