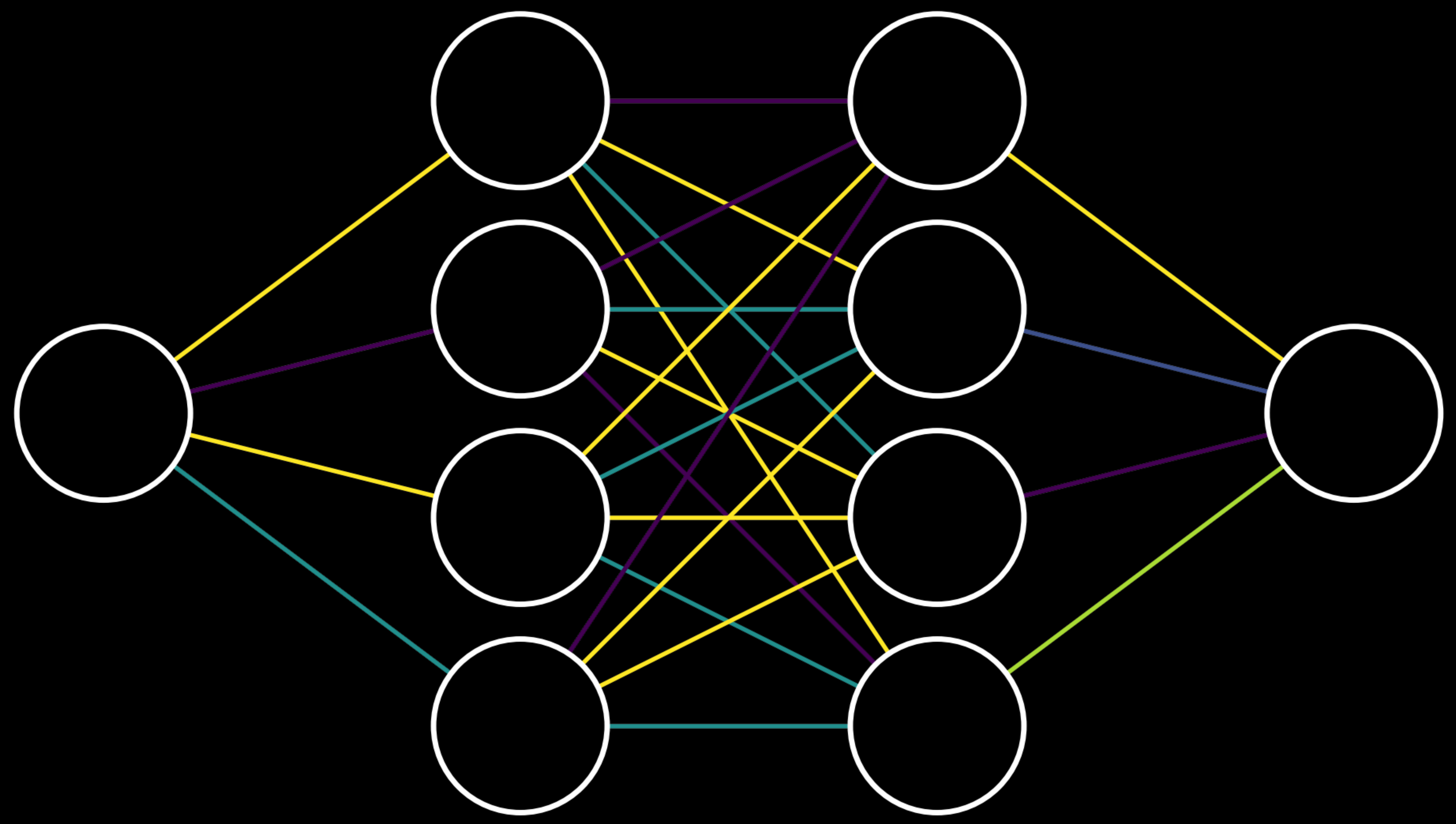
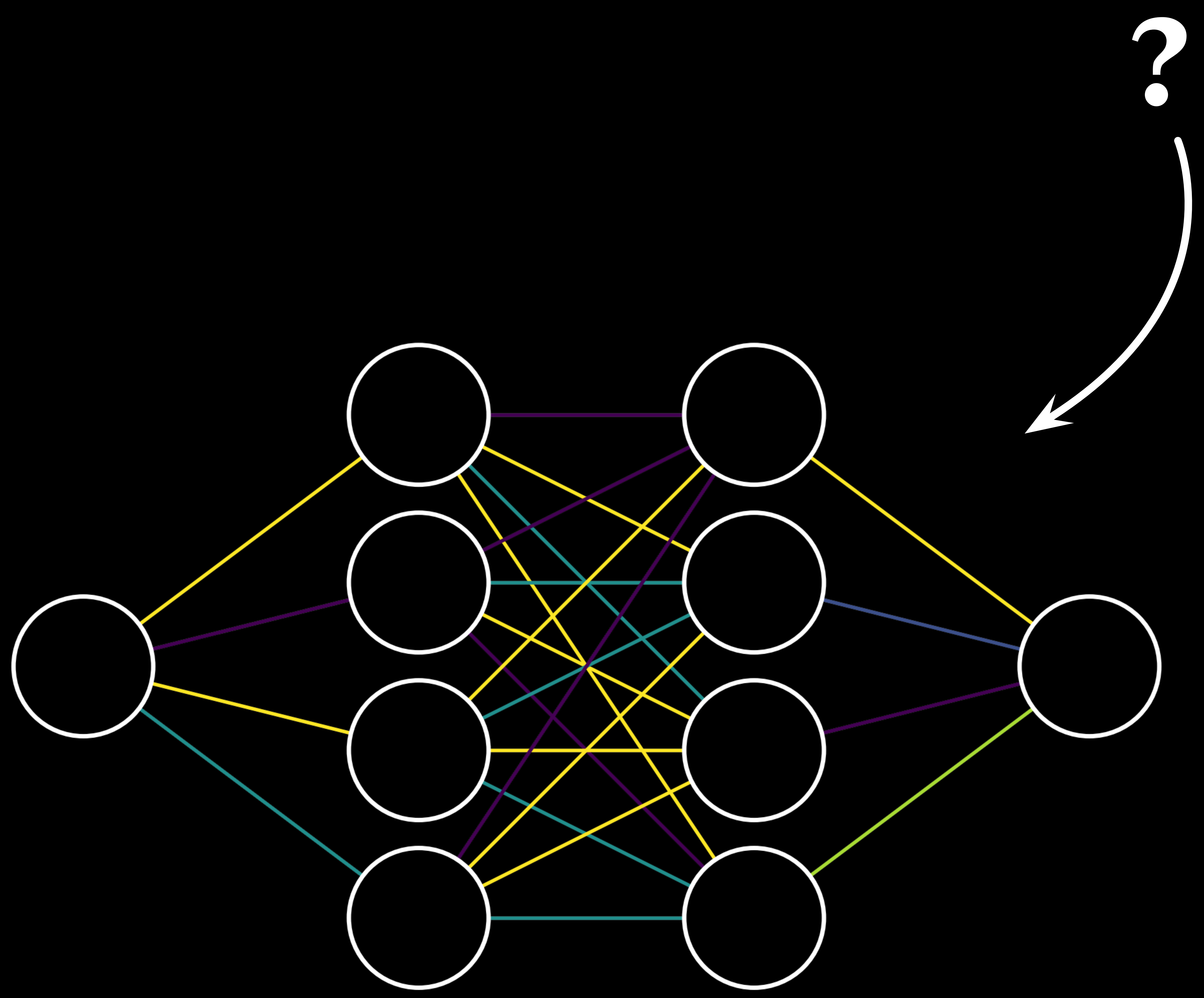
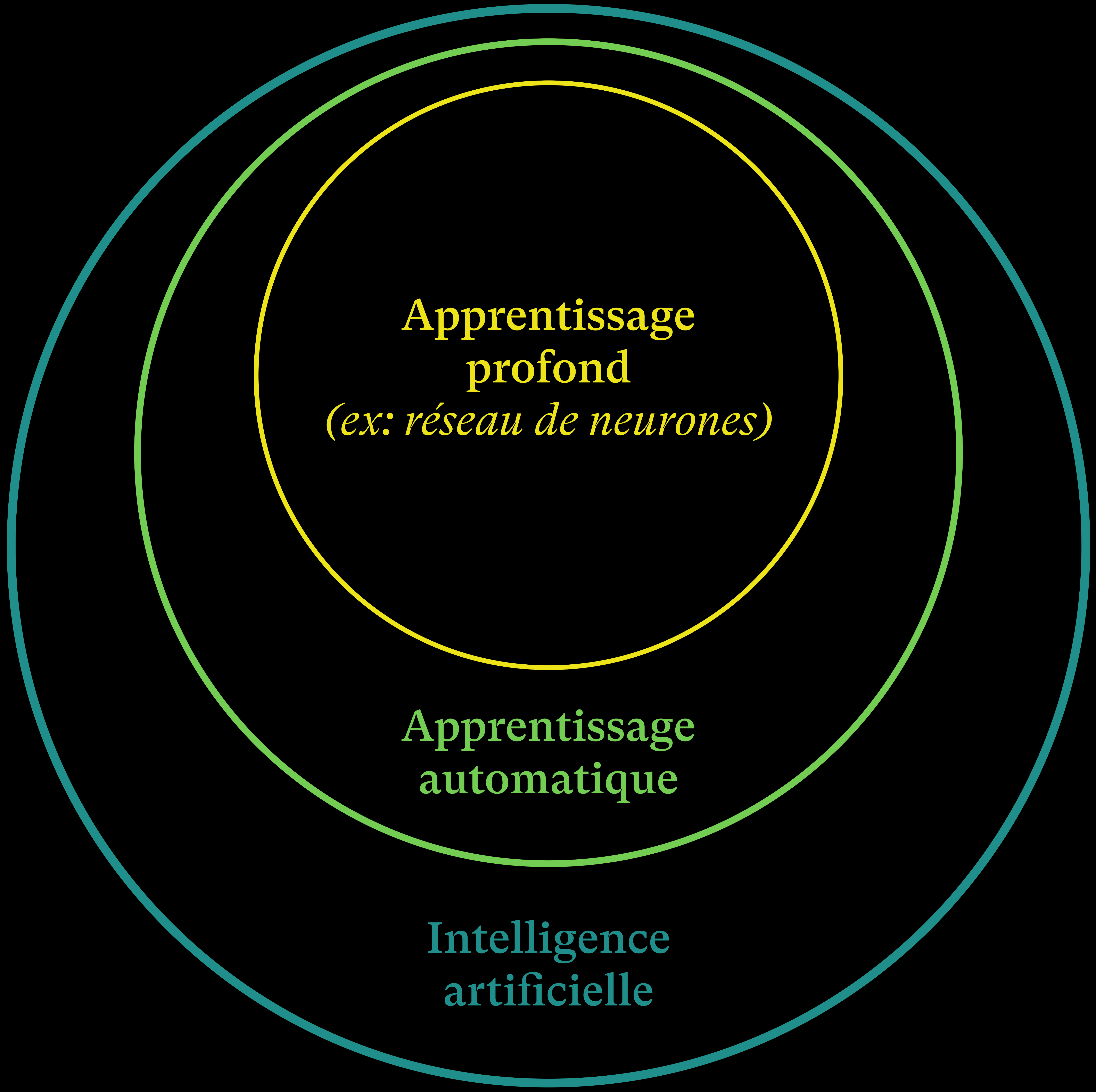
Introduction aux réseaux de neurones

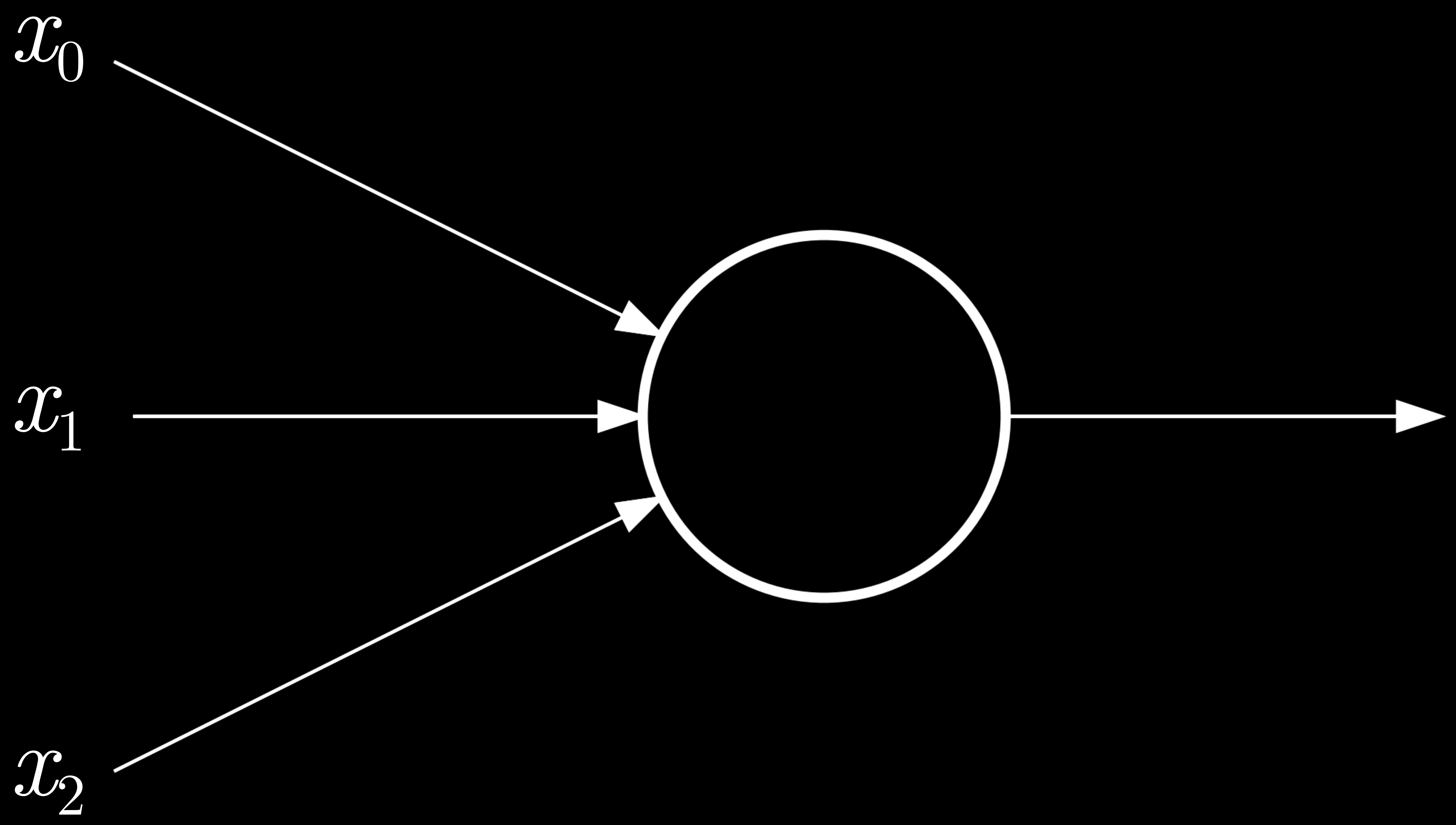
BOIN Tristan | JARRY Elisa

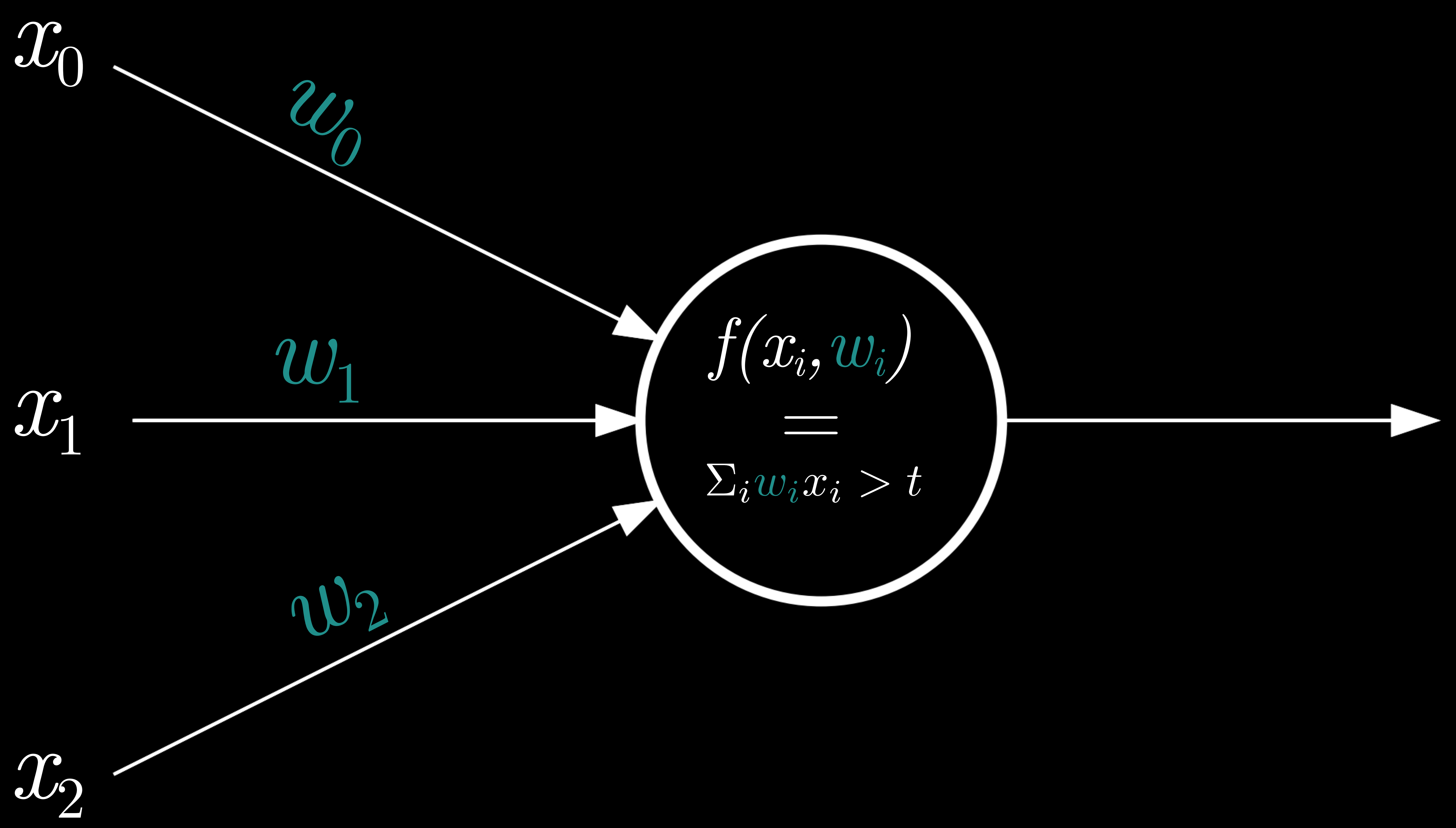




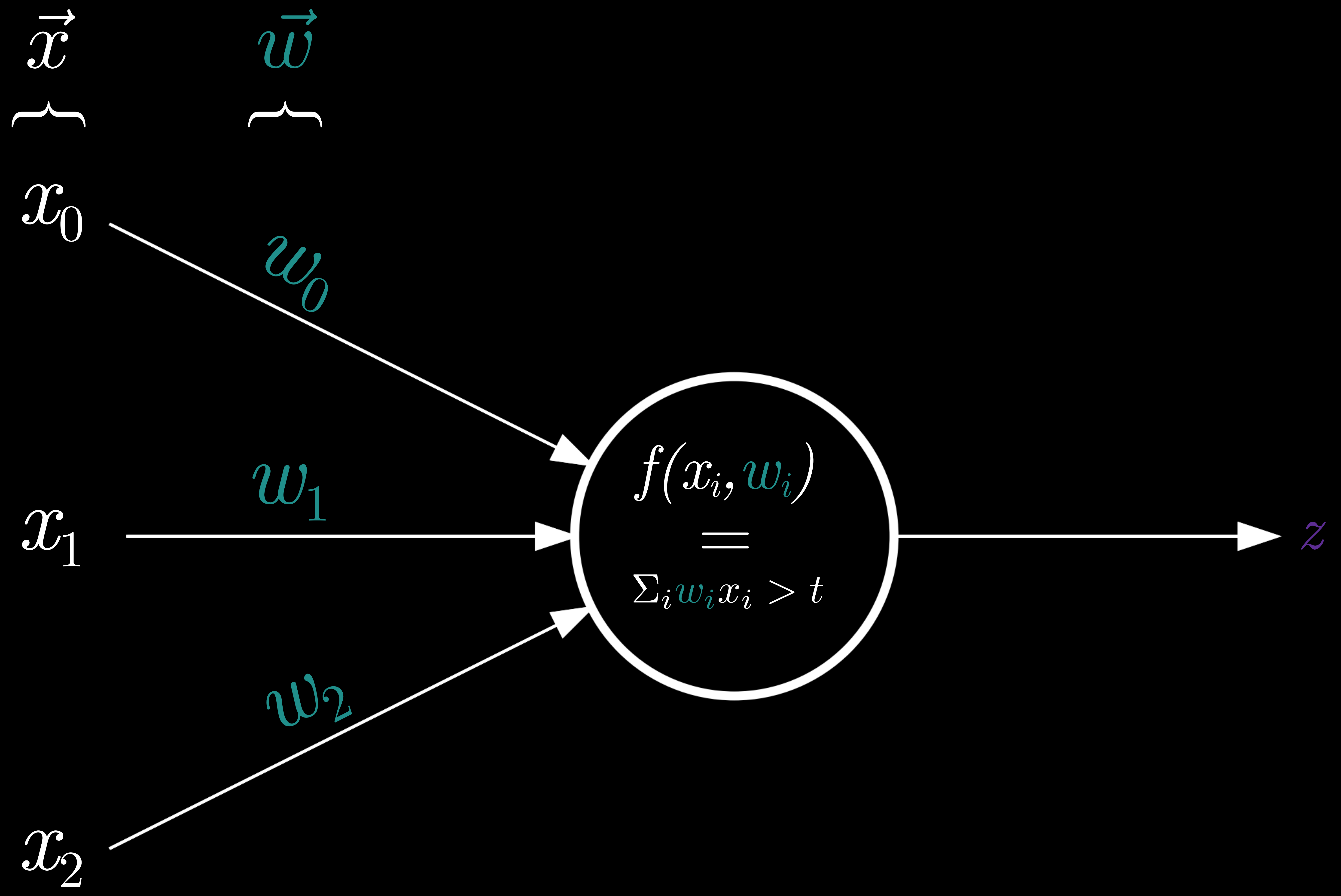

Apprentissage profond
(ex: réseau de neurones)
Apprentissage automatique
Intelligence artificielle


8 7 2 6 9 1 7 9 8 3 8 7 2 6 9 1
8 5 6 0 3 7 2 6 3 1 8 5 6 0 3 7
3 4 2 3 5 0 2 7 4 1 3 4 2 3 5 0
6 3 1 6 9 2 3 4 8 2 6 3 1 6 9 2
0 0 7 5 1 0 3 9 6 5 0 0 7 5 1 0
8 2 6 3 6 7 4 0 2 7 8 2 6 3 6 7
8 7 2 6 9 1 7 9 8 3 8 7 2 6 9 1
8 5 6 0 3 7 2 6 3 1 8 5 6 0 3 7

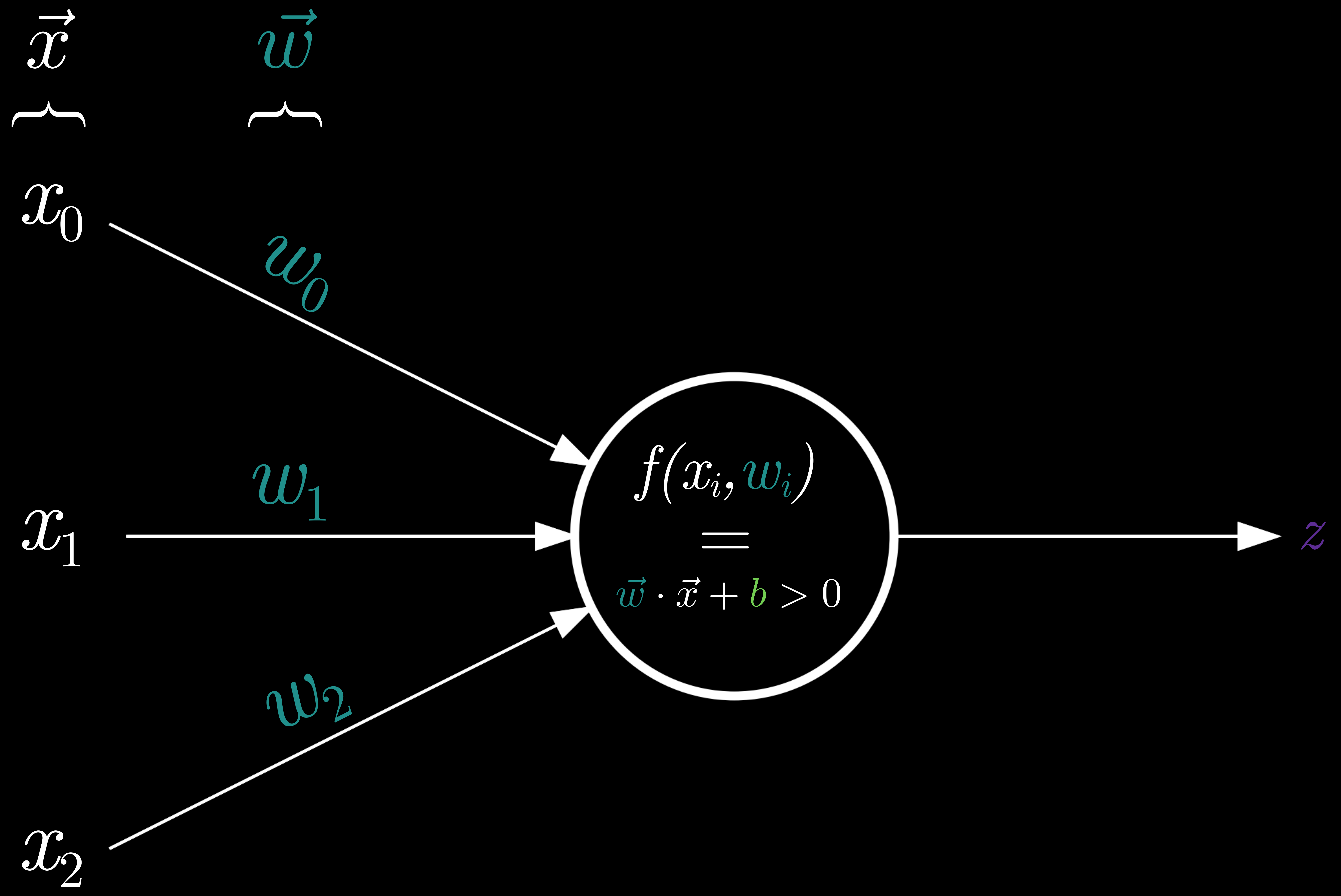

x_0
x_1
x_2



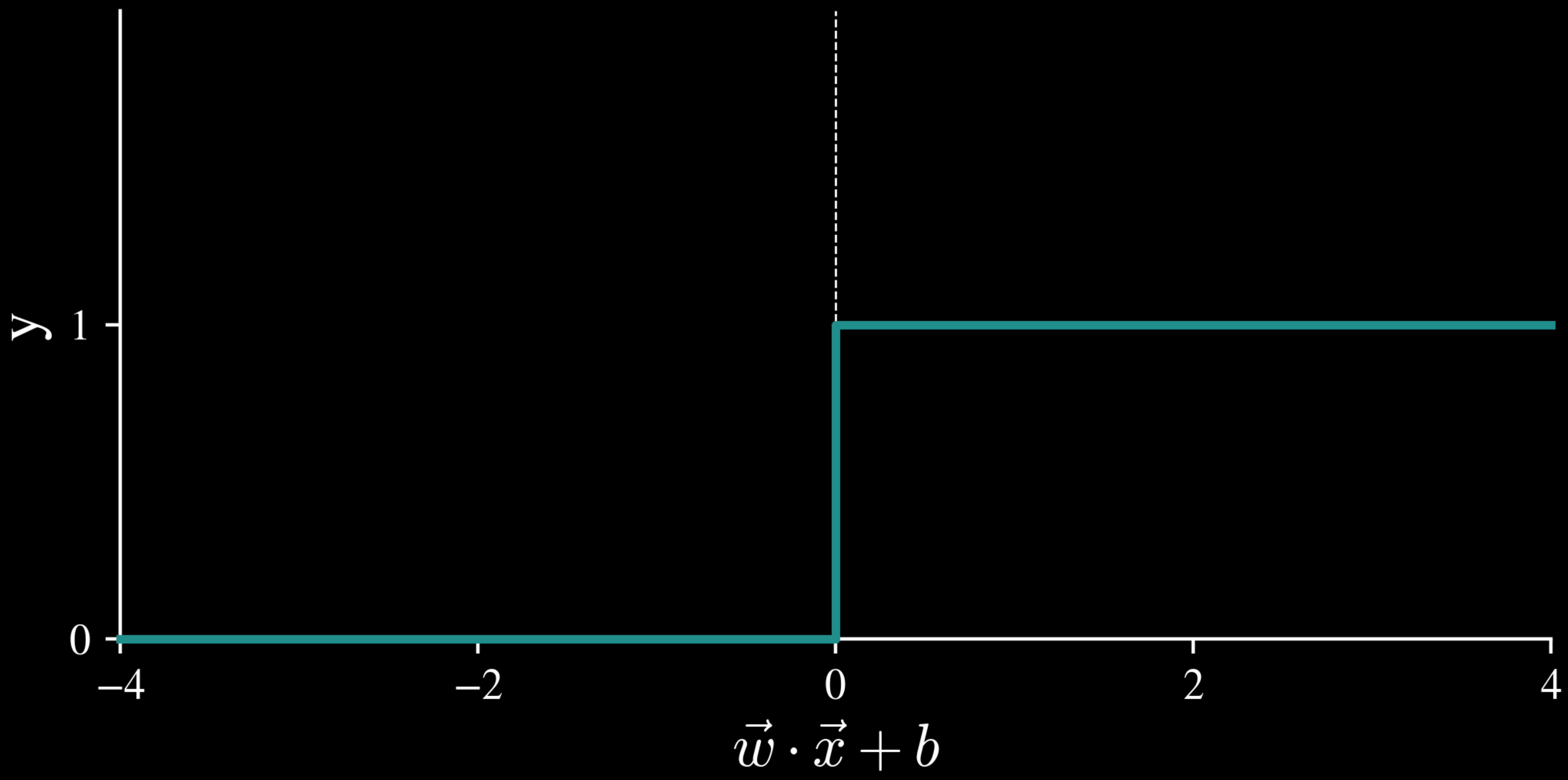
x_0
w_0
x_1
w_1
x_2
w_2
$f(x_i, w_i)$
$=$
$\Sigma_i w_i x_i > t$



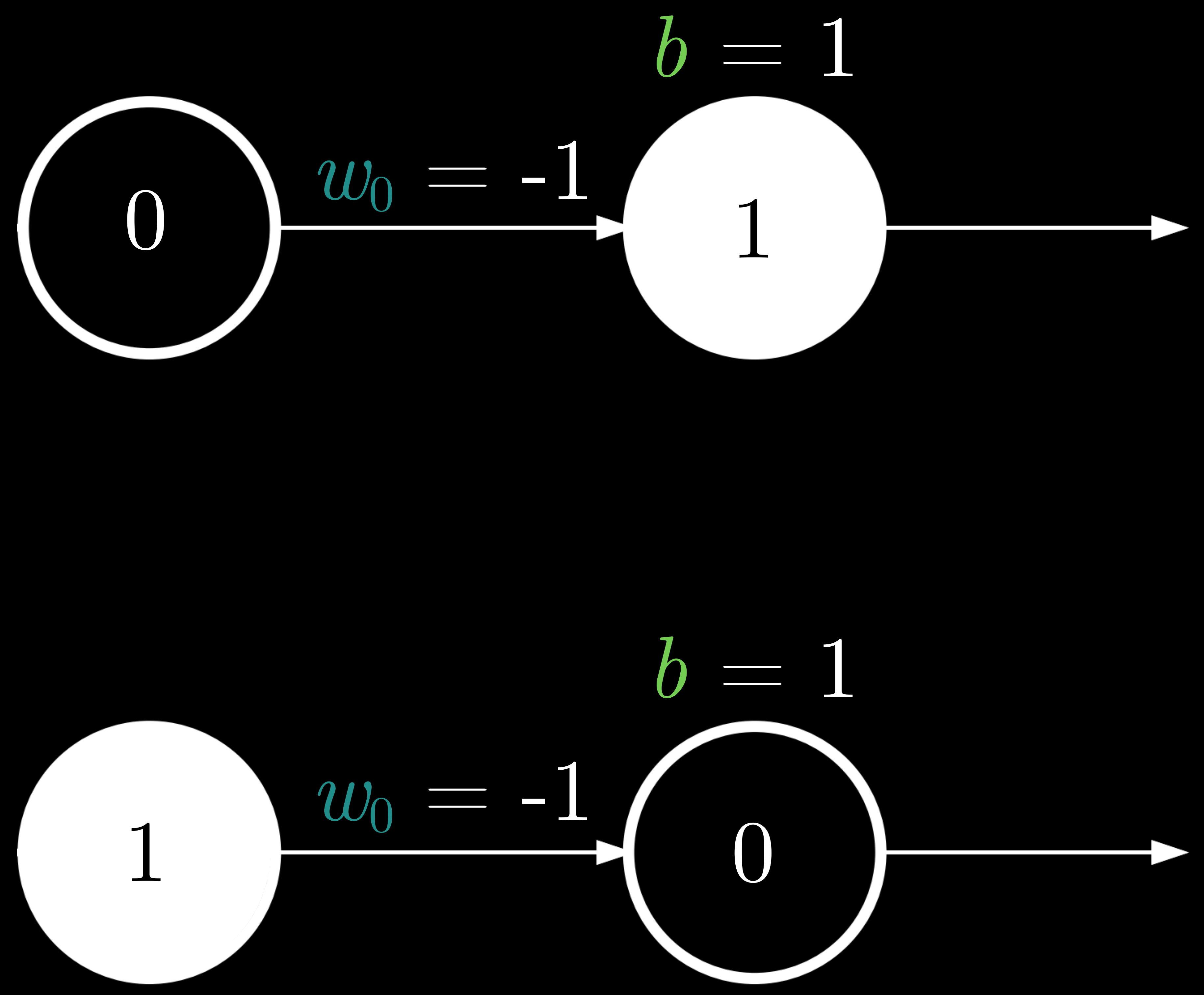
$\vec{x}$
$\vec{w}$
x_0
w_0
x_1
w_1
x_2
w_2
$f(x_i, w_i) = \Sigma_i w_i x_i > t$
z

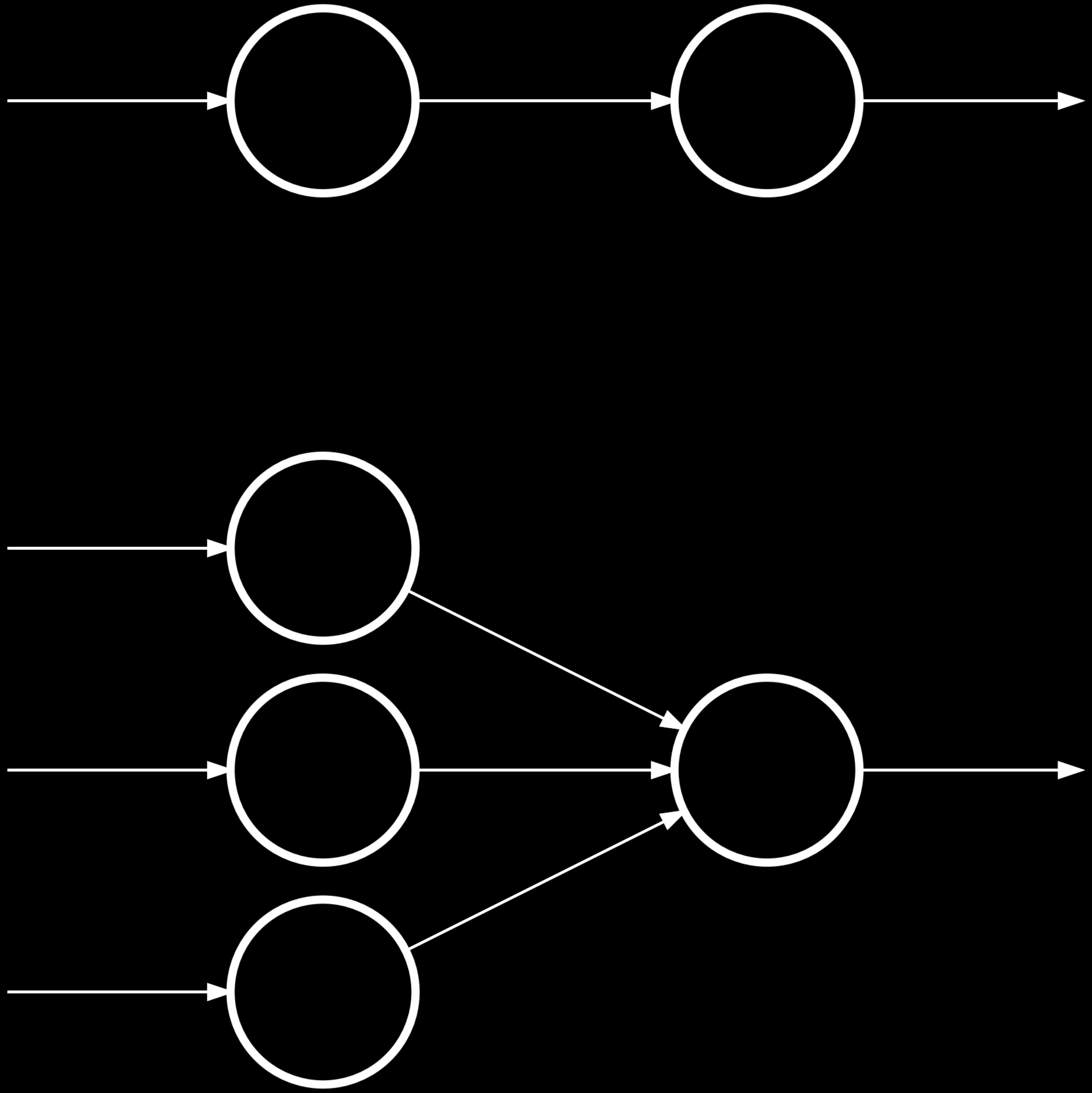

$\vec{x}$
$\vec{w}$
x_0
w_0
x_1
w_1
x_2
w_2
$f(x_i, w_i)$
$=$
$\vec{w} \cdot \vec{x} + b > 0$
z

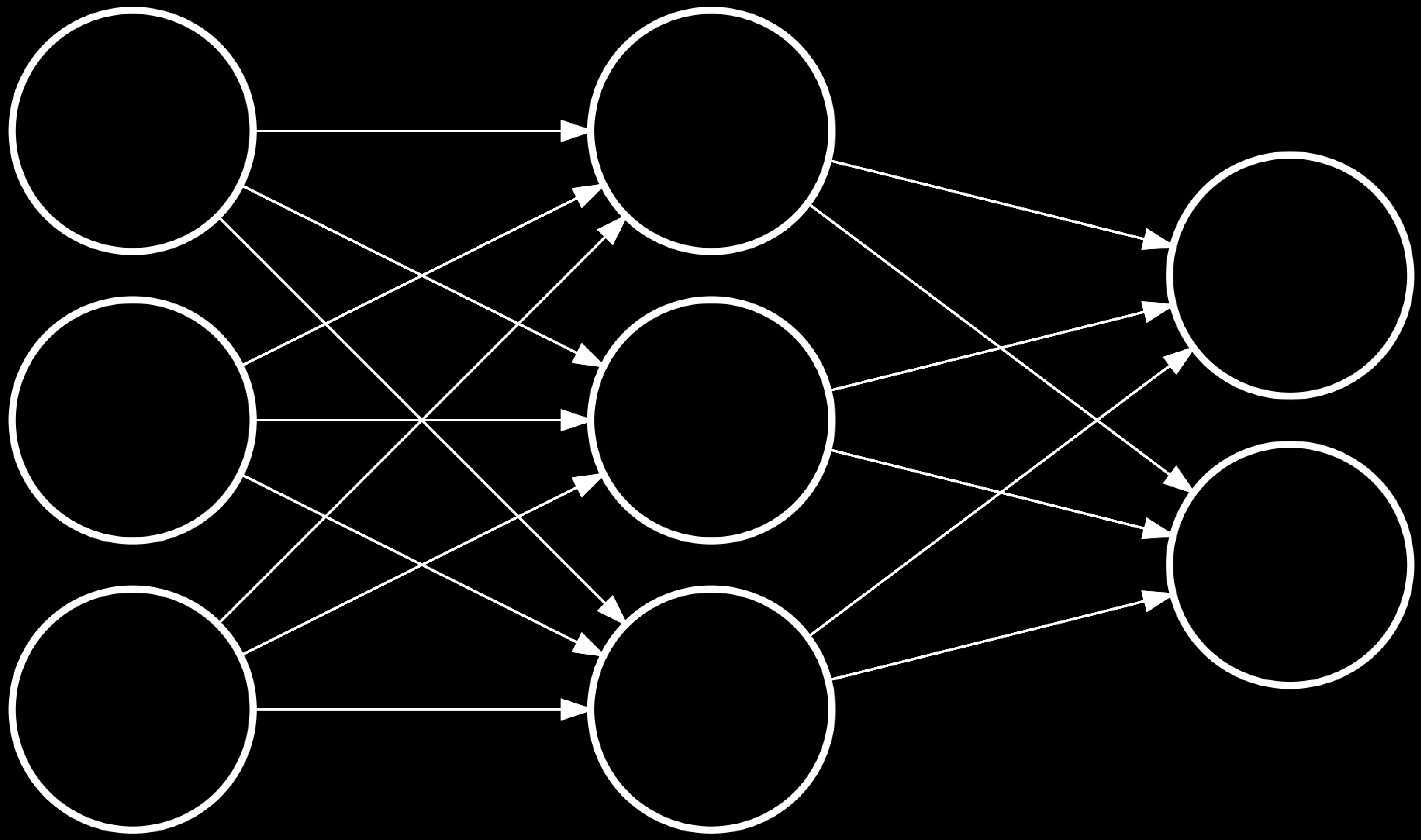
$$a = \begin{cases} 1 & \text{si } z > 0 \\ 0 & \text{sinon} \end{cases}$$

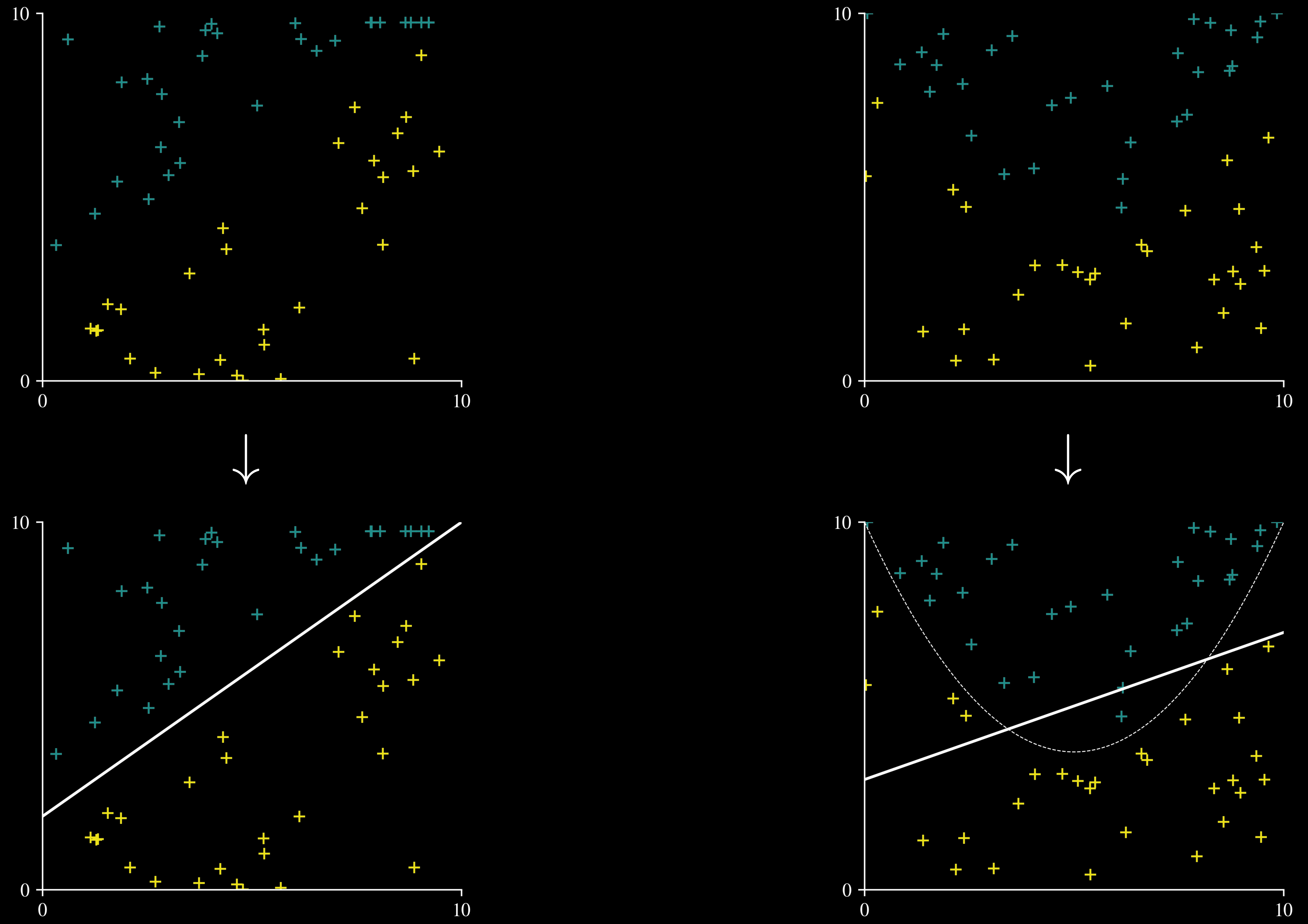


Inverseur :








10
0
0
10
10
0
0
10
10
0
0
10
10
0
0
10

$$z = \vec{w} \cdot \vec{x} + b$$

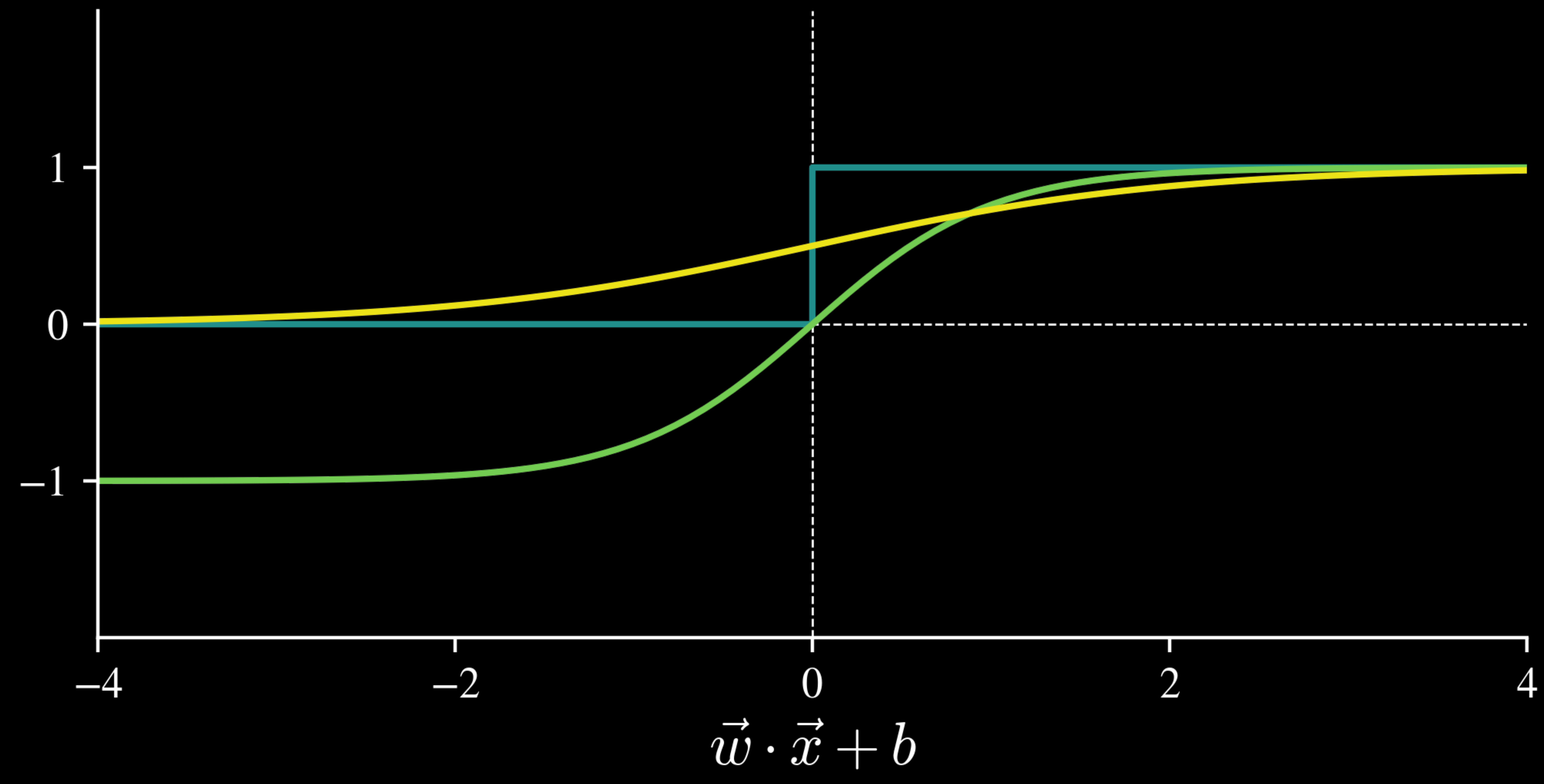
$$a = \begin{cases} 1 & \text{si } z > 0 \\ 0 & \text{sinon} \end{cases}$$

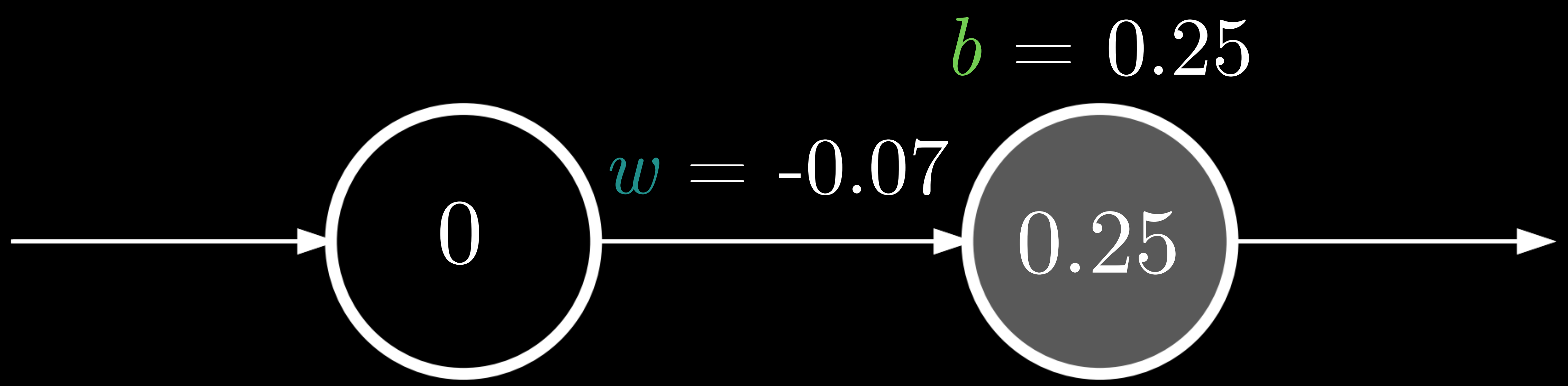
$\downarrow$

$$a = \tanh(z) = \frac{\sinh(z)}{\cosh(z)} = 1 - \frac{2}{e^{2z}+1}$$

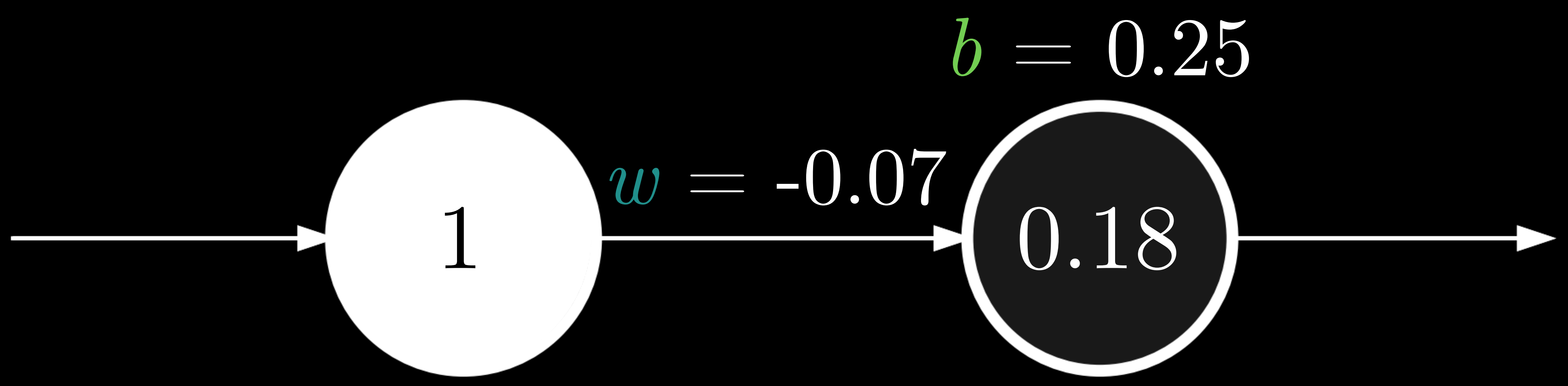
$\downarrow$

$$a = \sigma(z) = \frac{1}{2}(\tanh(\frac{z}{2}) + 1) = \frac{1}{1+e^{-z}}$$





$$C(x=0;w,b) = \frac{(0.25\text{-}1)^2}{2} = 0.28$$

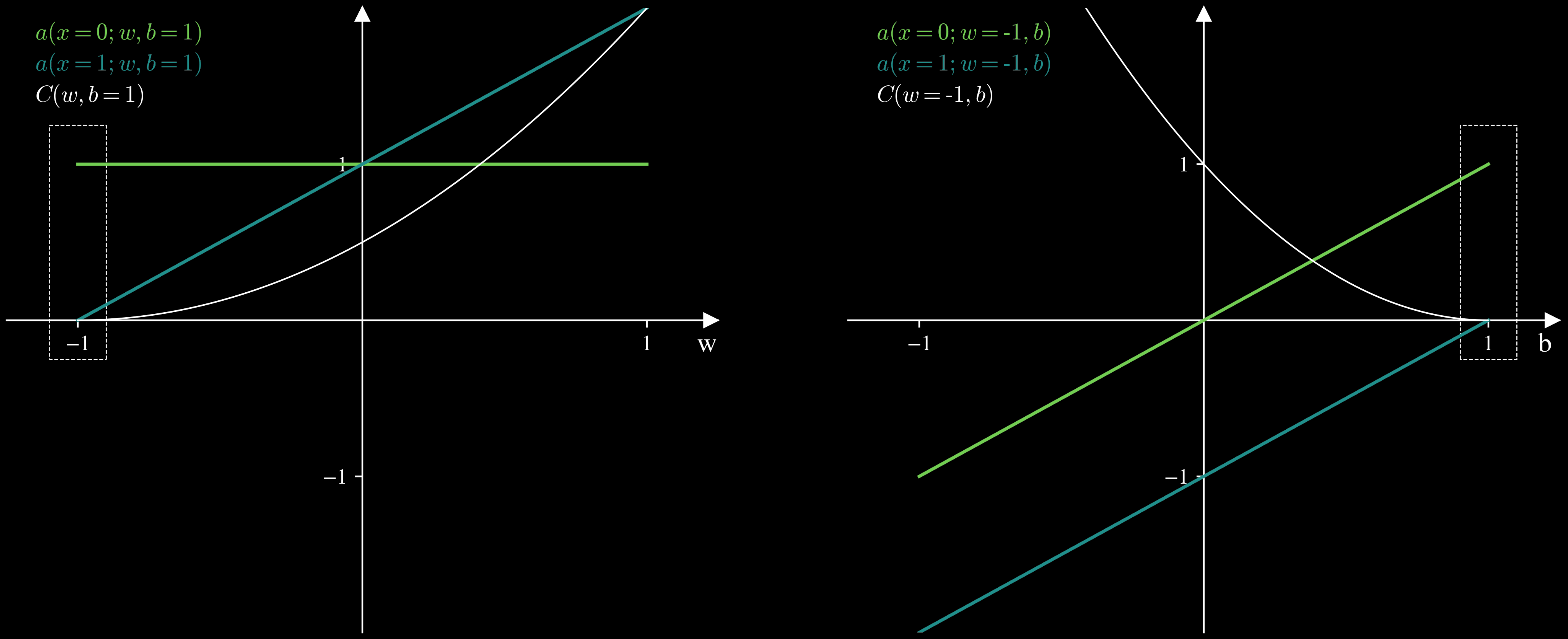


$$C(x=1;w,b) = \frac{(0.18\text{-}0)^2}{2} = 0.02$$

$$C(w,b) = \frac{(0.28+0.02)}{2} = 0.15$$

Erreur quadratique moyenne :

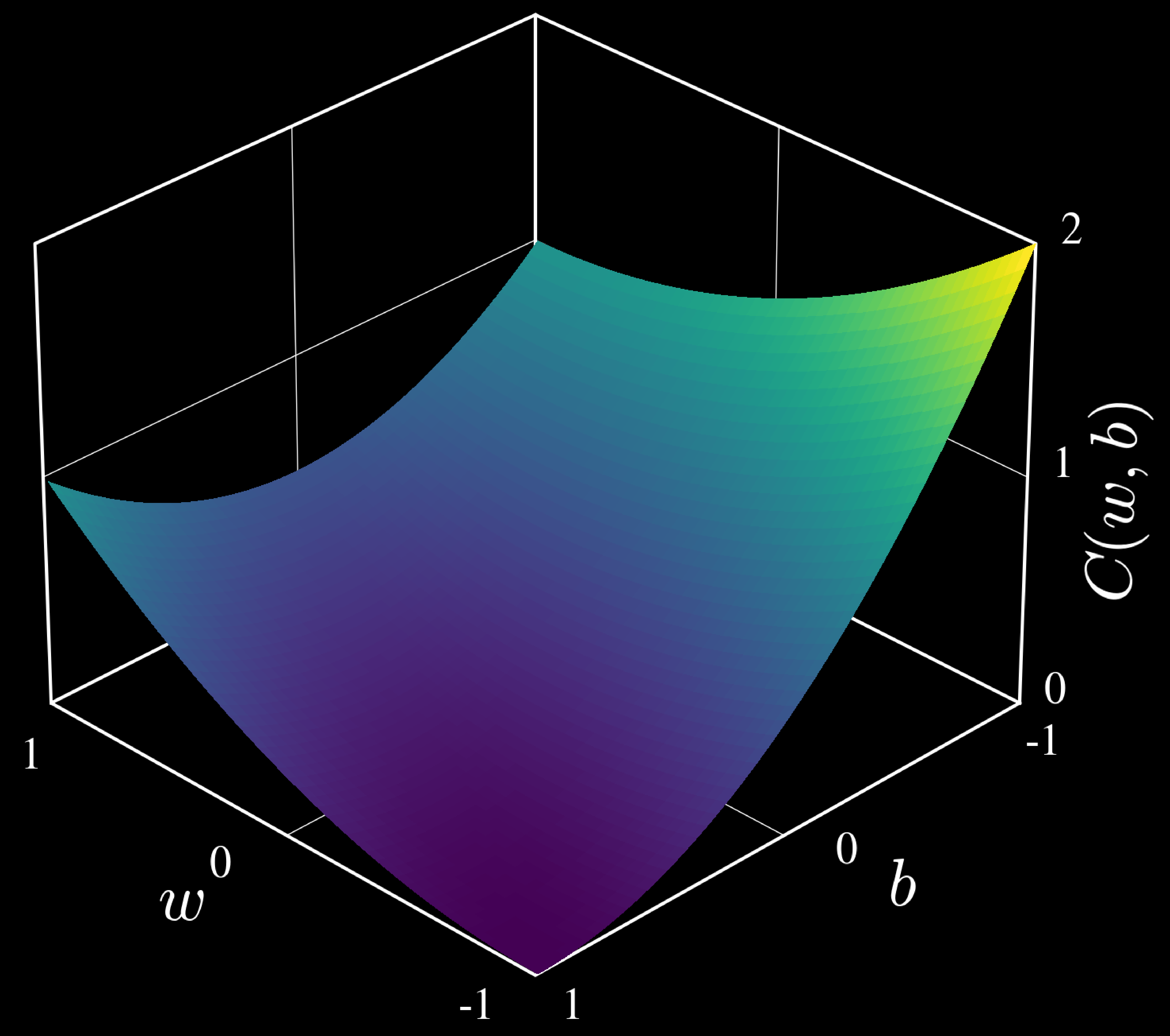
$$C(w,b) = \frac{1}{2n}\sum_x ||y(x) - a(w,b)||^2$$



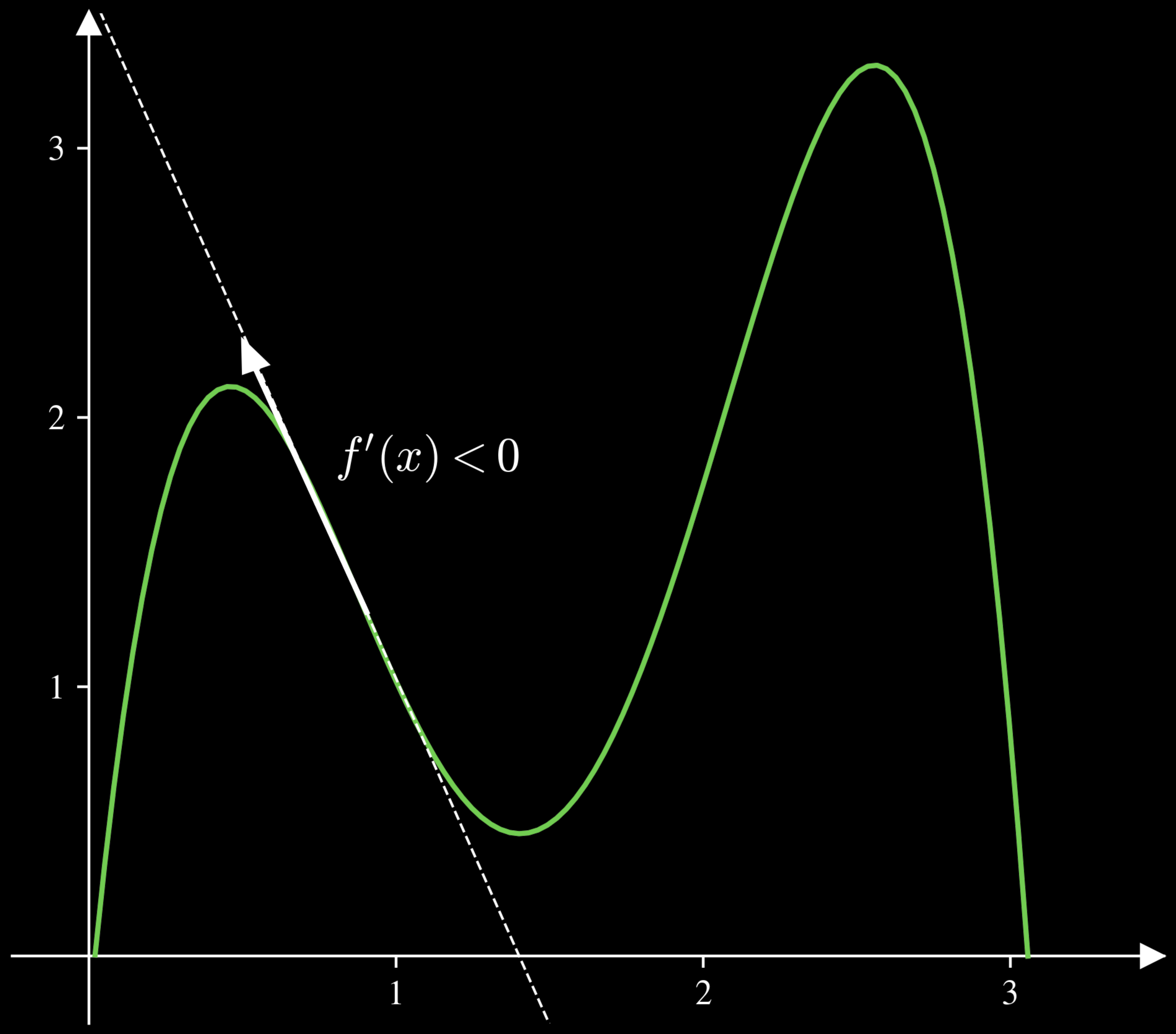
Entropie croisée :

$$C(w,b) = -\frac{1}{n}\sum_x [y(x)\ln(a(w,b)) + (1-y(x))\ln(1-a(w,b))]$$

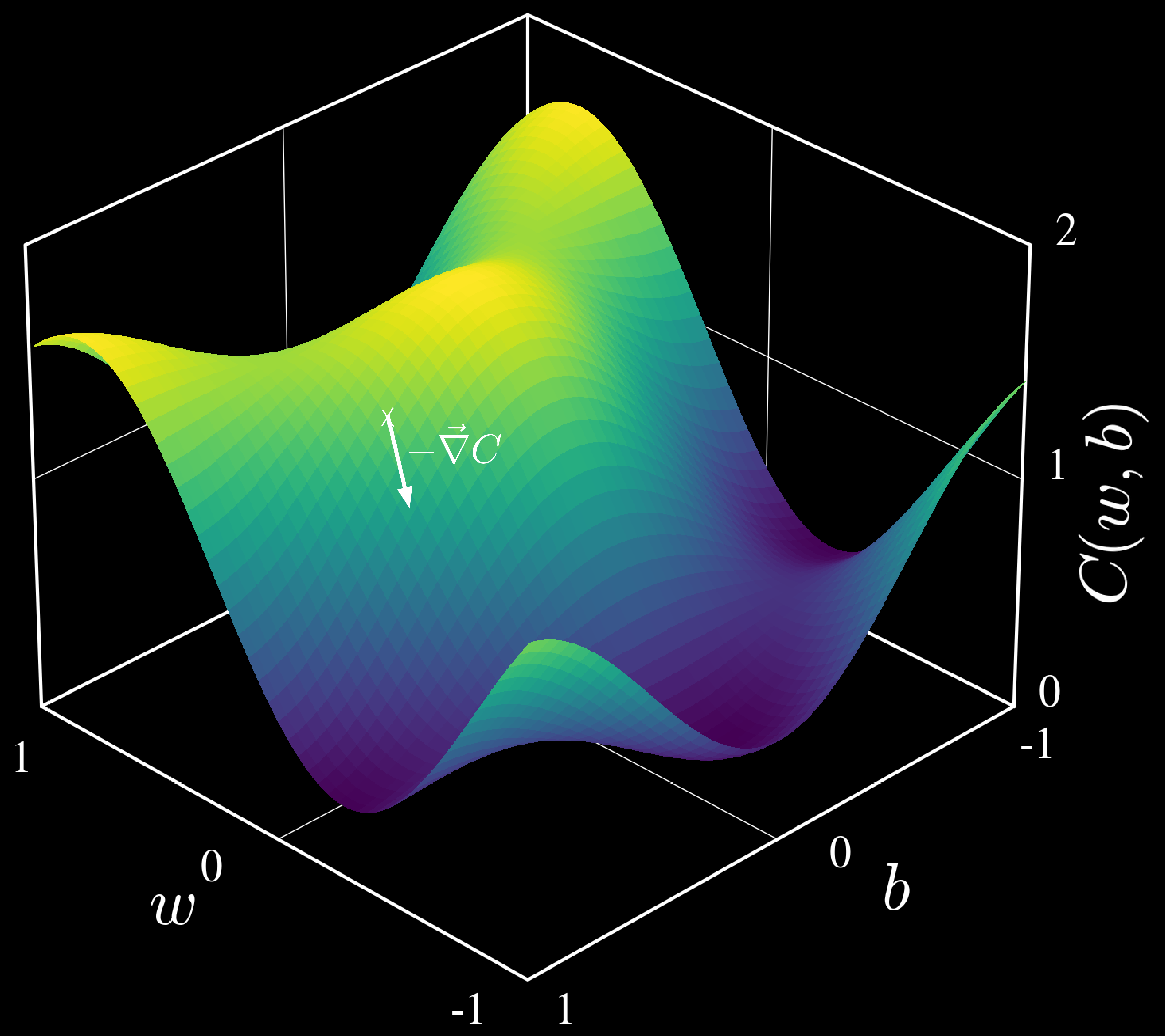
$$\begin{cases} \dfrac{\partial C}{\partial w_i} = \dfrac{\partial C}{\partial a} \cdot \dfrac{\partial a}{\partial w_i} = -\dfrac{1}{n}\displaystyle\sum_x \left[\dfrac{y(x)}{a} - \dfrac{(1-y(x))}{(1-a)}\right] \cdot \dfrac{\partial a}{\partial w_i} = \dfrac{1}{n}\displaystyle\sum_x x_i(a-y(x)) \\[2em] \dfrac{\partial C}{\partial b} = \dfrac{\partial C}{\partial a} \cdot \dfrac{\partial a}{\partial b} = -\dfrac{1}{n}\displaystyle\sum_x \left[\dfrac{y(x)}{a} - \dfrac{(1-y(x))}{(1-a)}\right] \cdot \dfrac{\partial a}{\partial b} = \dfrac{1}{n}\displaystyle\sum_x (a-y(x)) \end{cases}$$



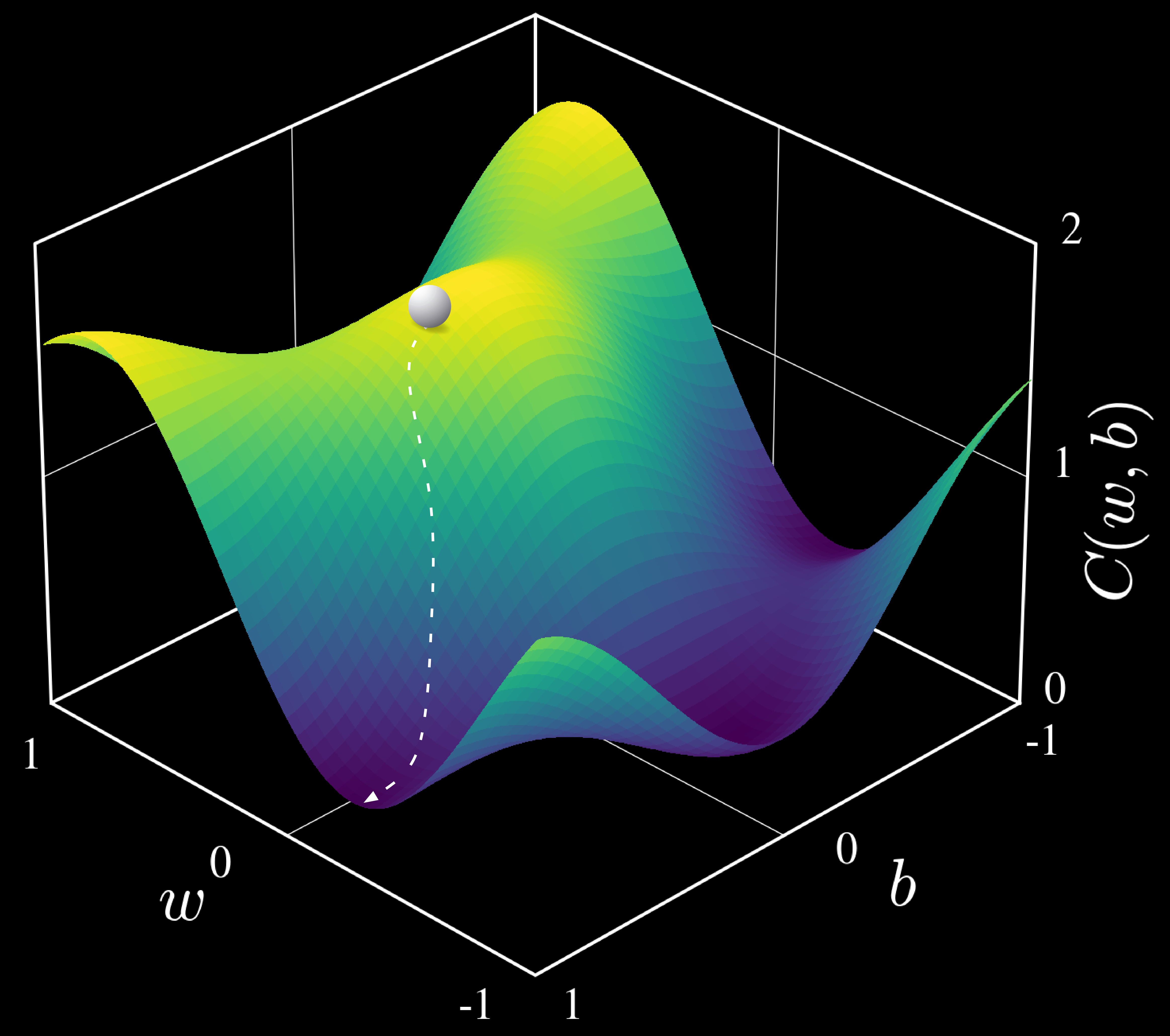
2
1
0
$C(w, b)$
1
0
w
-1
1
0
b
-1



3
2
1
1
2
3
$f'(x) < 0$



$-\vec{\nabla}C$
2
1
0
$C(w, b)$
1
0
-1
w
-1
1
0
b



2
1
0
$C(w, b)$
1
0
-1
w
1
0
-1
b

Pour l'erreur quadratique moyenne :

$$\vec{\nabla} C = \begin{bmatrix} \frac{\partial C}{\partial w^0_{0,0}} \\ \frac{\partial C}{\partial b^0_0} \\ \cdots \\ \frac{\partial C}{\partial w^L_{n,n}} \\ \frac{\partial C}{\partial b^L_n} \end{bmatrix}$$

$$\frac{\partial C}{\partial w^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial w^L}$$

$$\frac{\partial C}{\partial b^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial b^L}$$

$$\begin{cases} \frac{\partial a^L}{\partial z^L} = \sigma'(z^L) \\ \frac{\partial z^L}{\partial w^L} = a^{L-1} \\ \frac{\partial z^L}{\partial b^L} = 1 \end{cases}$$

soit : $\frac{\partial C}{\partial w^L} = \frac{\partial C}{\partial a^L} \cdot \sigma'(z^L) a^{L-1}$ et $\frac{\partial C}{\partial b^L} = \frac{\partial C}{\partial a^L} \cdot \sigma'(z^L)$

$$\frac{\partial C}{\partial a^L} = \textbf{?}$$

$$\frac{\partial C}{\partial a^L}$$

$\swarrow$ $\searrow$

Pour la couche de sortie :

$$2(a^L - y(x))$$

Pour les couches cachées :

$$\frac{\partial C}{\partial a^{L+1}} \cdot \frac{\partial a^{L+1}}{\partial z^{L+1}} \cdot \frac{\partial z^{L+1}}{\partial a^L} \begin{cases} \frac{\partial C}{\partial a^{L+1}} = 2(a^{L+1} - y(x)) \\ \frac{\partial a^{L+1}}{\partial z^{L+1}} = \sigma'(z^{L+1}) \\ \frac{\partial z^{L+1}}{\partial a^L} = w^{L+1} \end{cases}$$

$$\frac{\partial C}{\partial w^L} = \frac{\partial C}{\partial a^L} \cdot \sigma'(z^L) a^{L-1}$$

$$\frac{\partial C}{\partial b^L} = \frac{\partial C}{\partial a^L} \cdot \sigma'(z^L)$$

$$\frac{\partial C}{\partial w^L} \begin{cases} 2(a^L - y(x))\sigma'(z^L)a^{L-1} \\ 2(a^{L+1} - y(x))\sigma'(z^{L+1})w^{L+1}\sigma'(z^L)a^{L-1} \end{cases}$$

$$\frac{\partial C}{\partial b^L} \begin{cases} 2(a^L - y(x))\sigma'(z^L) \\ 2(a^{L+1} - y(x))\sigma'(z^{L+1})w^{L+1}\sigma'(z^L) \end{cases}$$

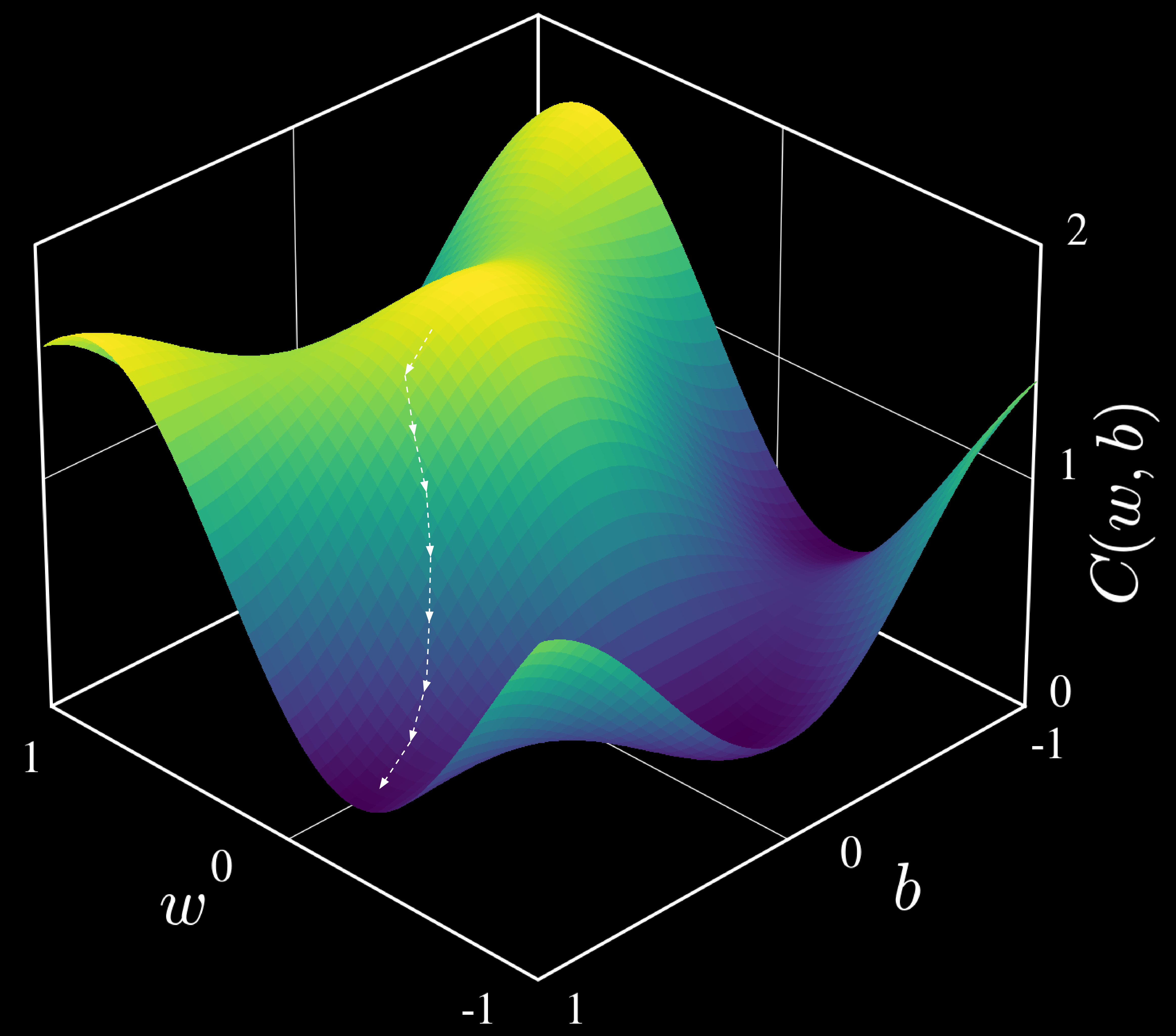
gradient moyen :

$$\vec{\nabla} C = \frac{1}{n} \sum_x \vec{\nabla} C_x$$

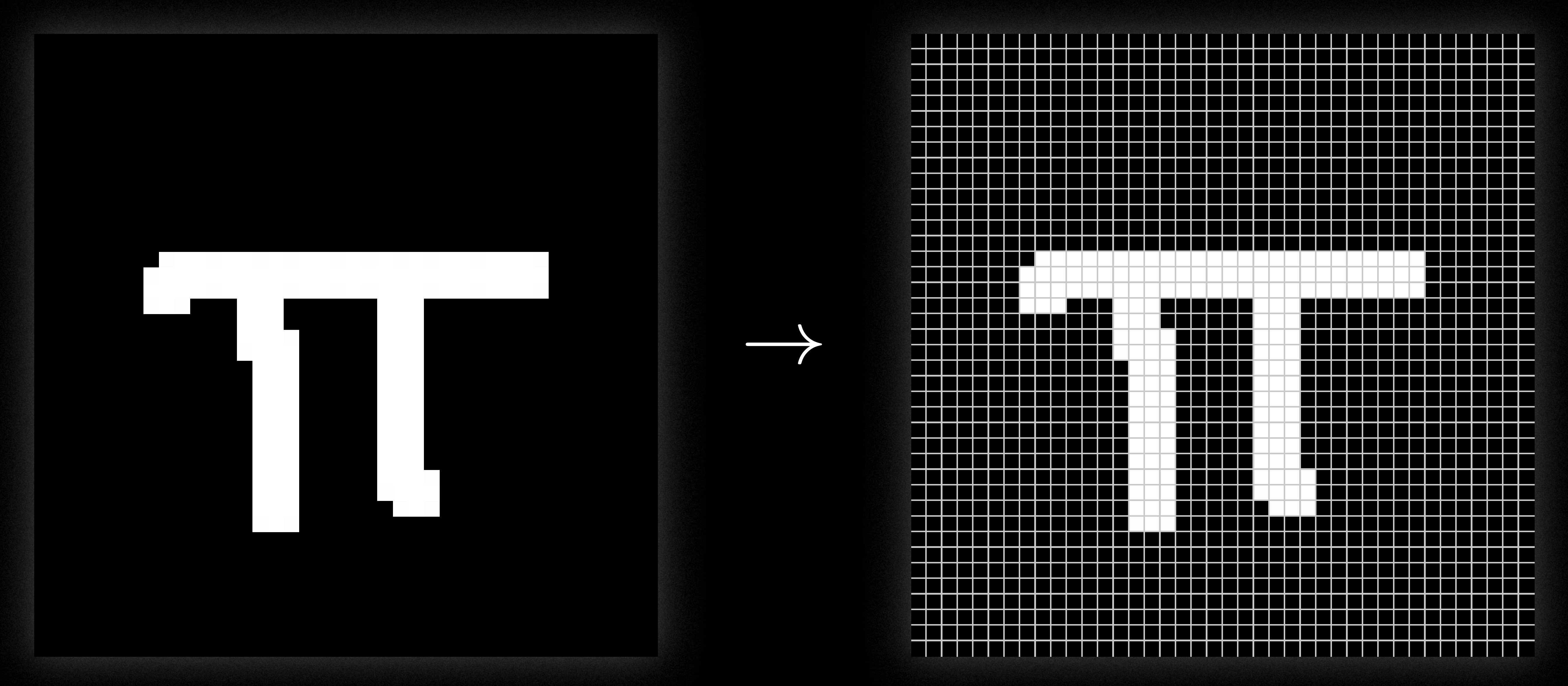
$$\begin{cases} w_{i,j}^L \rightarrow w_{i,j}^L + \Delta w_{i,j}^L \quad \text{avec} \quad \Delta w_{i,j}^L = -\eta \frac{\partial C}{\partial w_{i,j}^L} \\ \\ b_{i,j}^L = b_{i,j}^L + \Delta b_{i,j}^L \quad \text{avec} \quad \Delta b_{i,j}^L = -\eta \frac{\partial C}{\partial b_{i,j}^L} \end{cases}$$

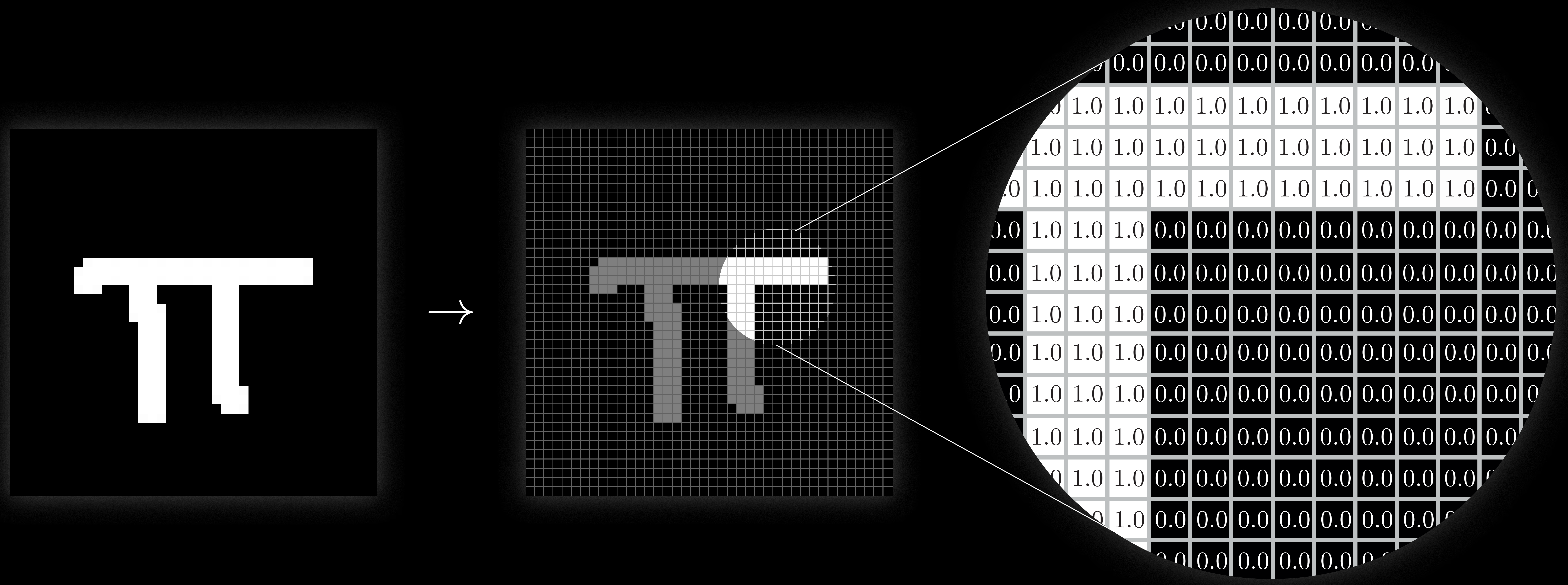
Les 4 équations de ~~Maxwell~~ rétropropagation

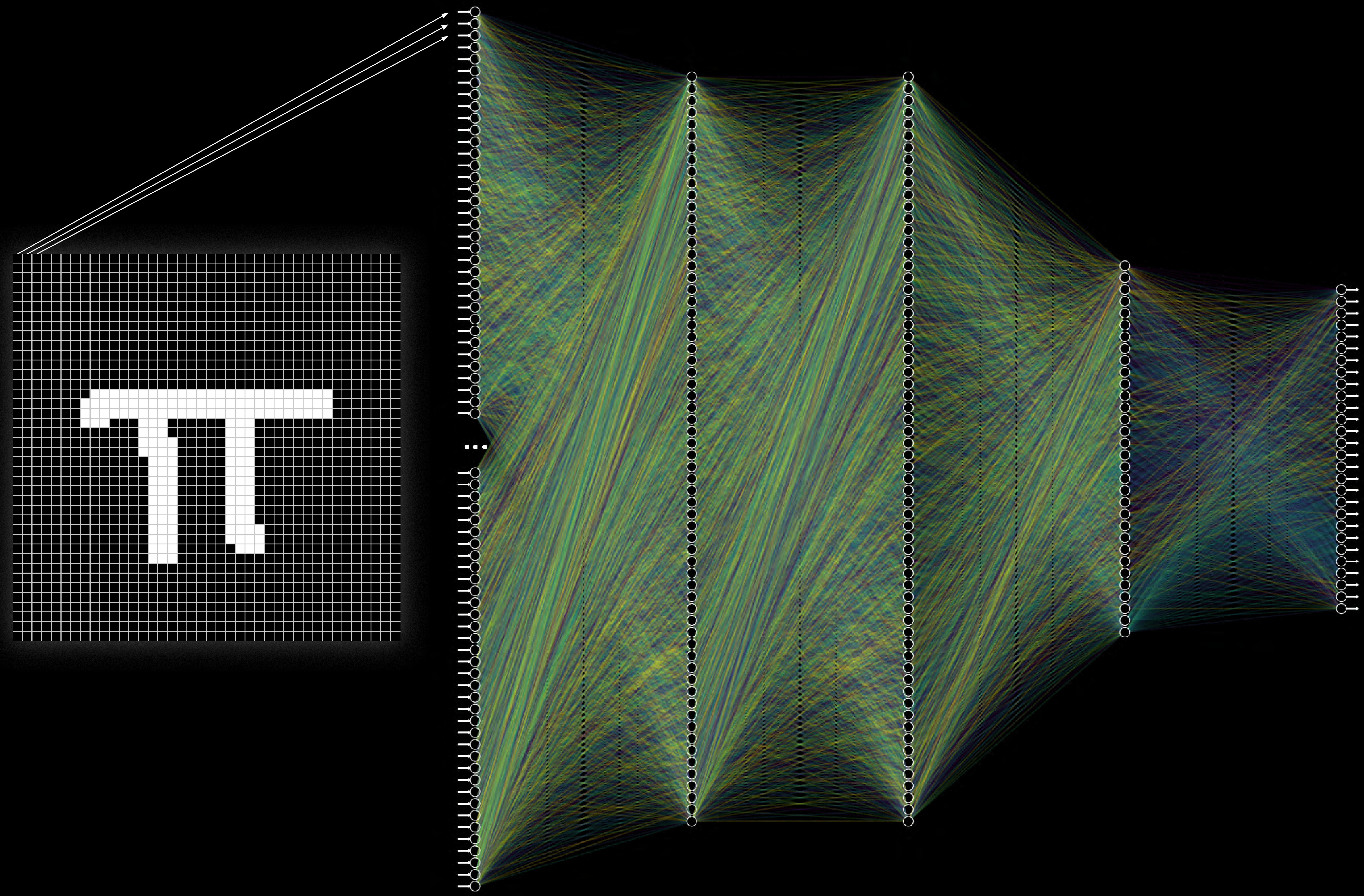
$$
\left[\begin{array}{lr}
\delta^L = \nabla_a C \odot \sigma'(z^L) & (1) \\[2em]
\delta^L = ((w^{L+1})^T \delta^{L+1}) \odot \sigma'(z^L) & (2) \\[2em]
\dfrac{\partial C}{\partial b_j^L} = \delta_j^L & (3) \\[2em]
\dfrac{\partial C}{\partial w_j^L} = a^{L-1} \delta_j^L & (4)
\end{array}\right]
$$

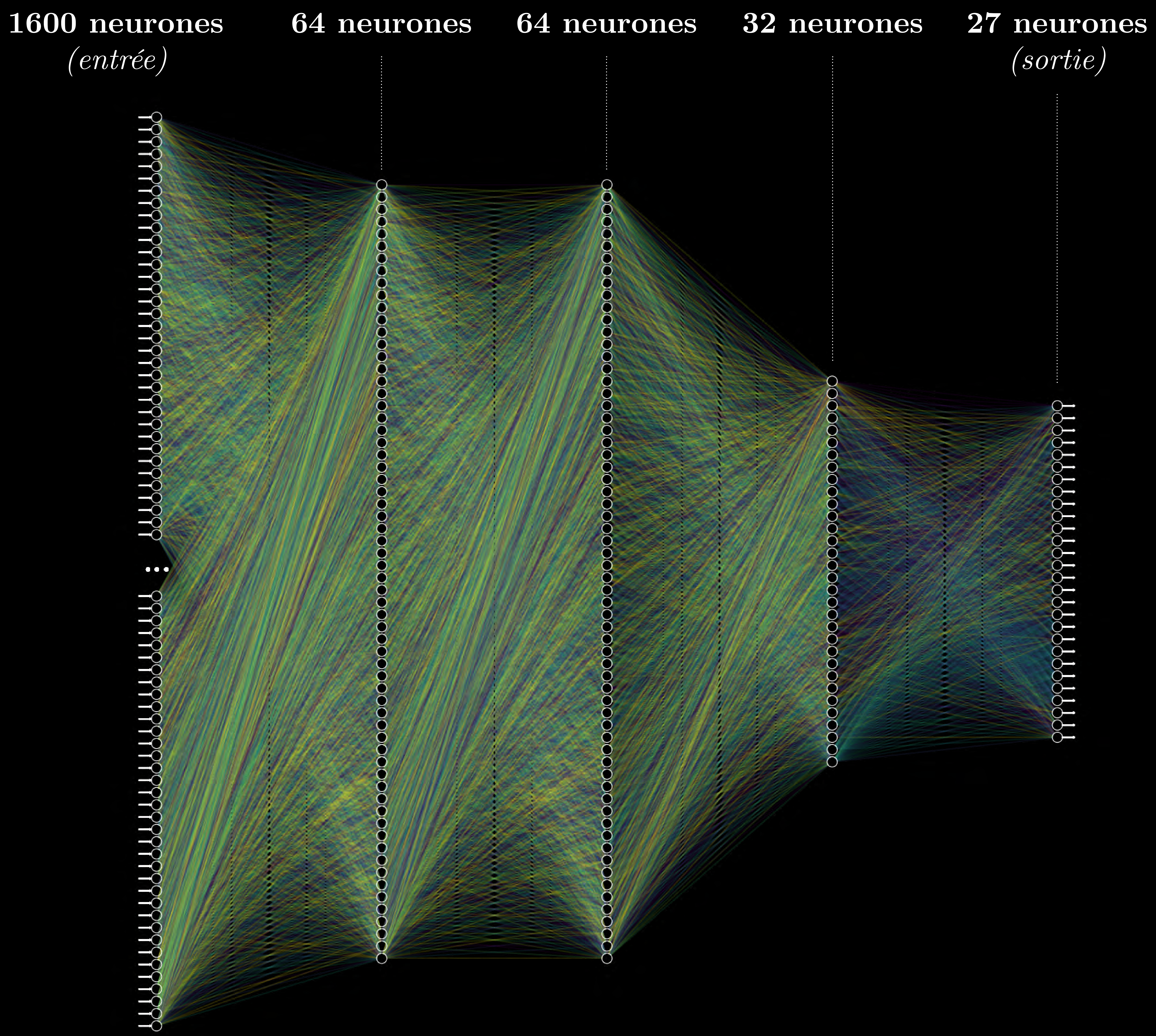


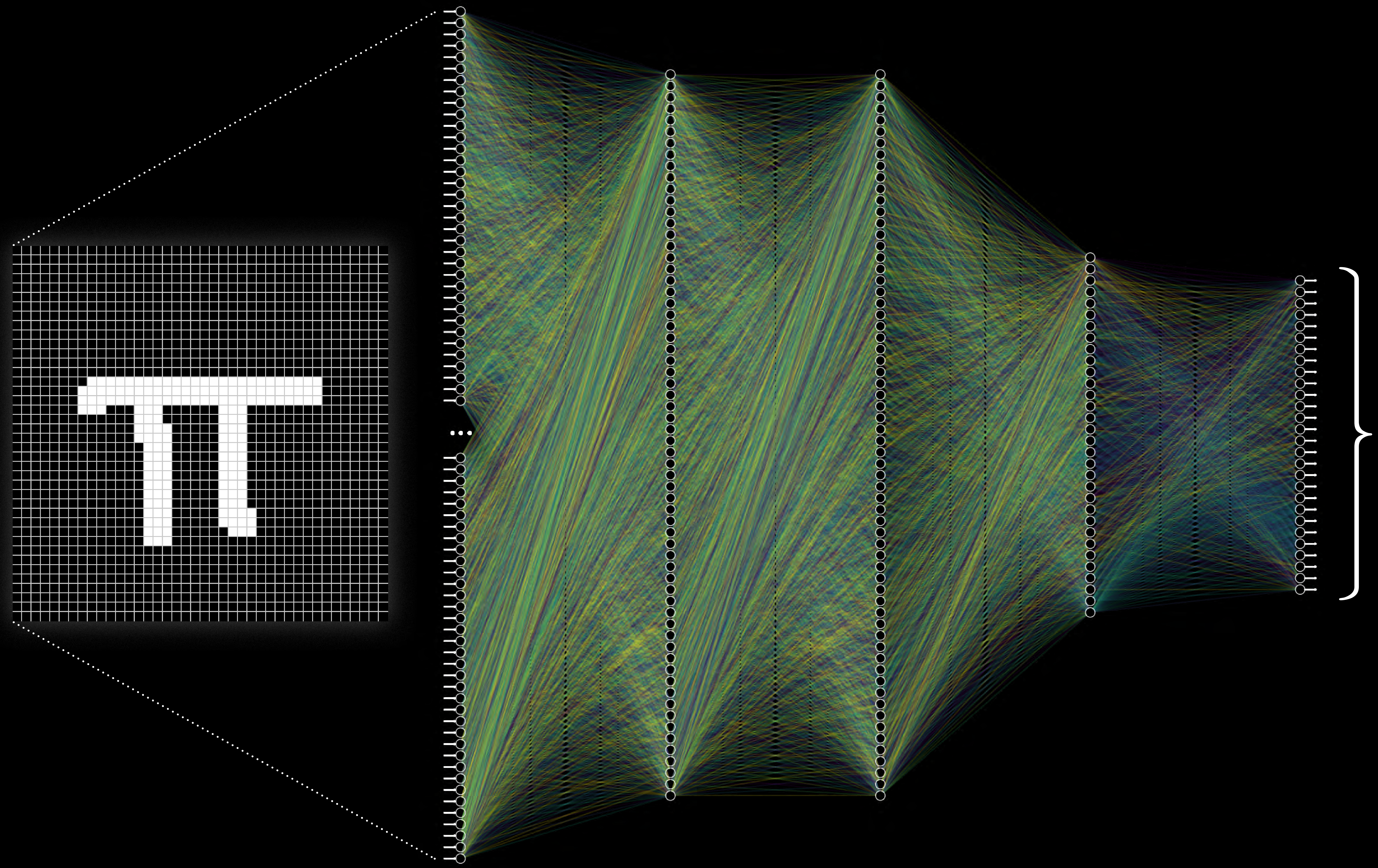
2
1
0
$C(w,b)$
1
0
w
-1
1
0
b
-1

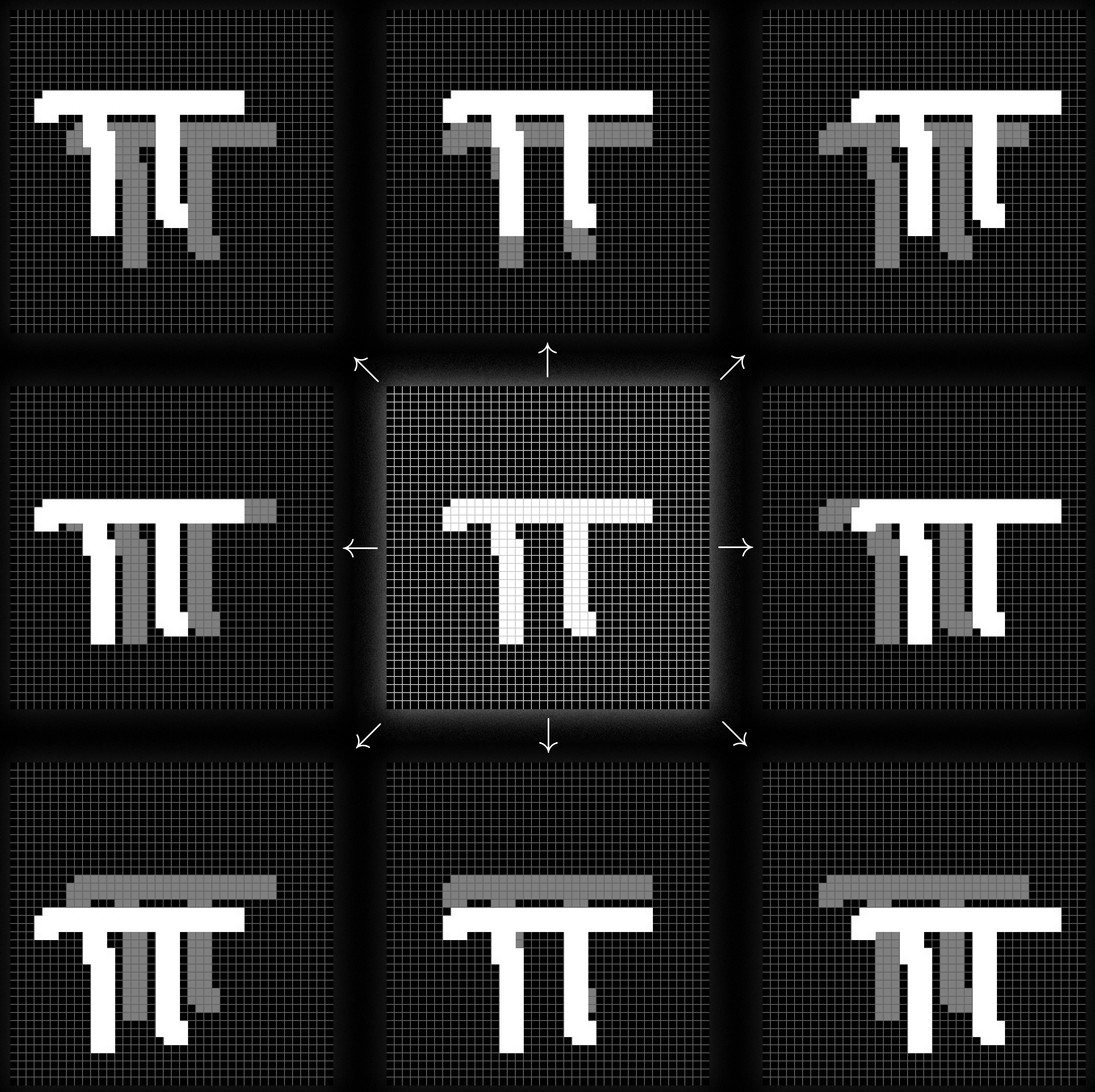



0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.0
1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0



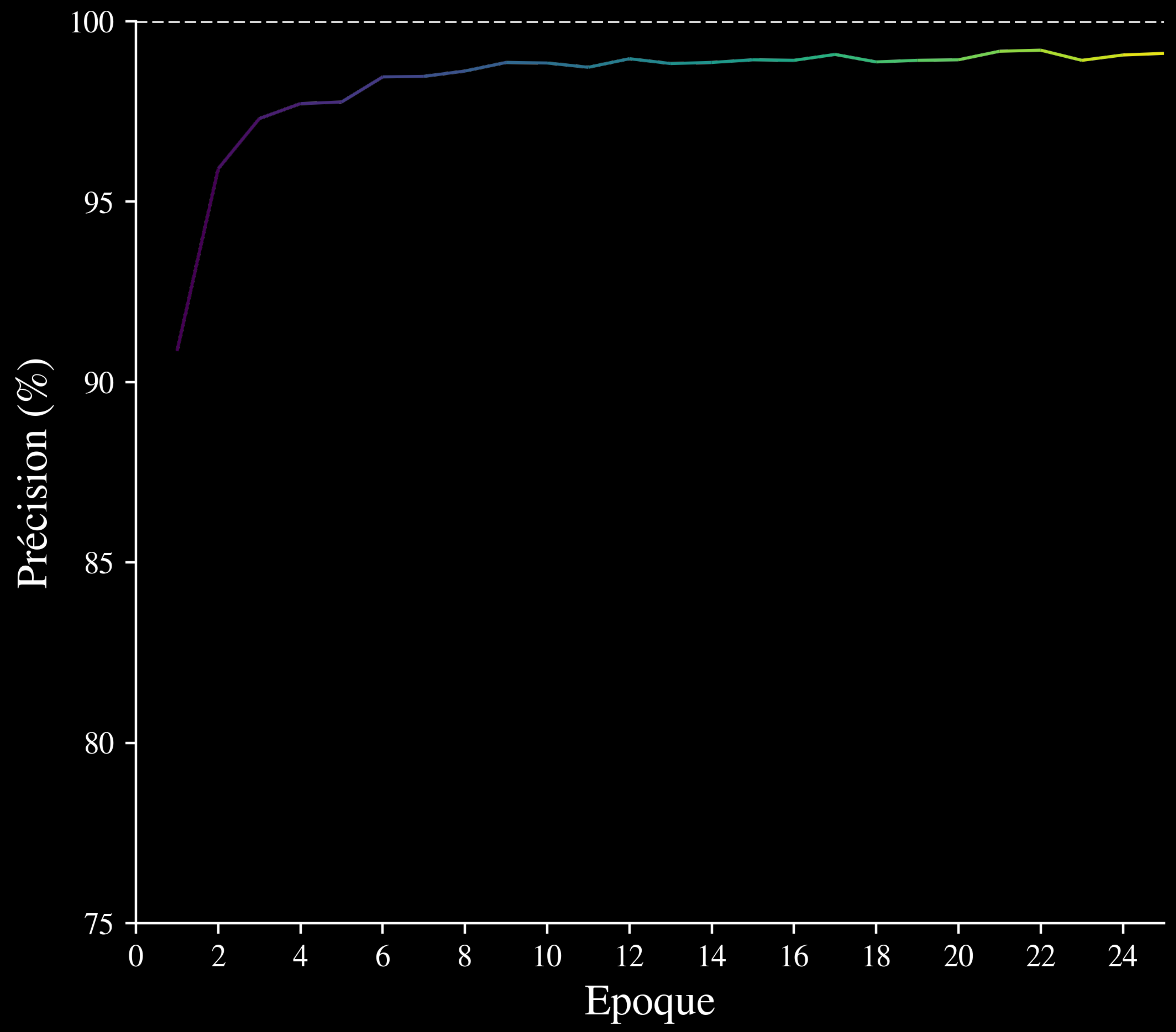

1600 neurones
(entrée)
64 neurones
64 neurones
32 neurones
27 neurones
(sortie)


...
symbole le plus probable :
"π"



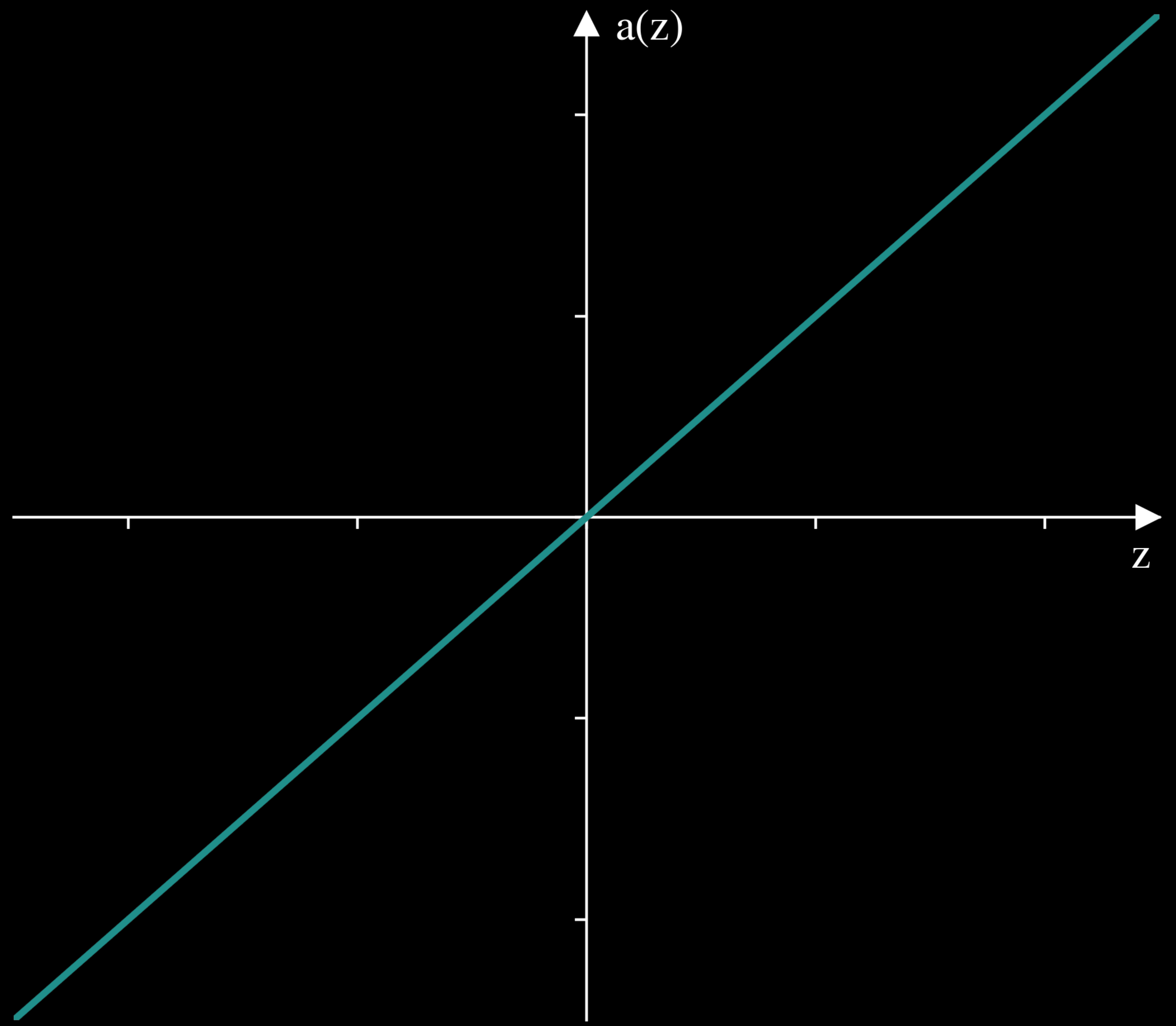
Statistiques du réseau

Nombre d'images de training	378 126
Taille totale des images compressées	$\simeq$ 36 Mo
Temps de training	$\simeq$ 30 min
Proportion de caractérisations correctes	99.20 %
Temps de réponse pour 100 images	9.3 ms
Taille du réseau	$\simeq$ 2 Mo
Taux d'apprentissage η	0.10
Nombre d'époques de training	25



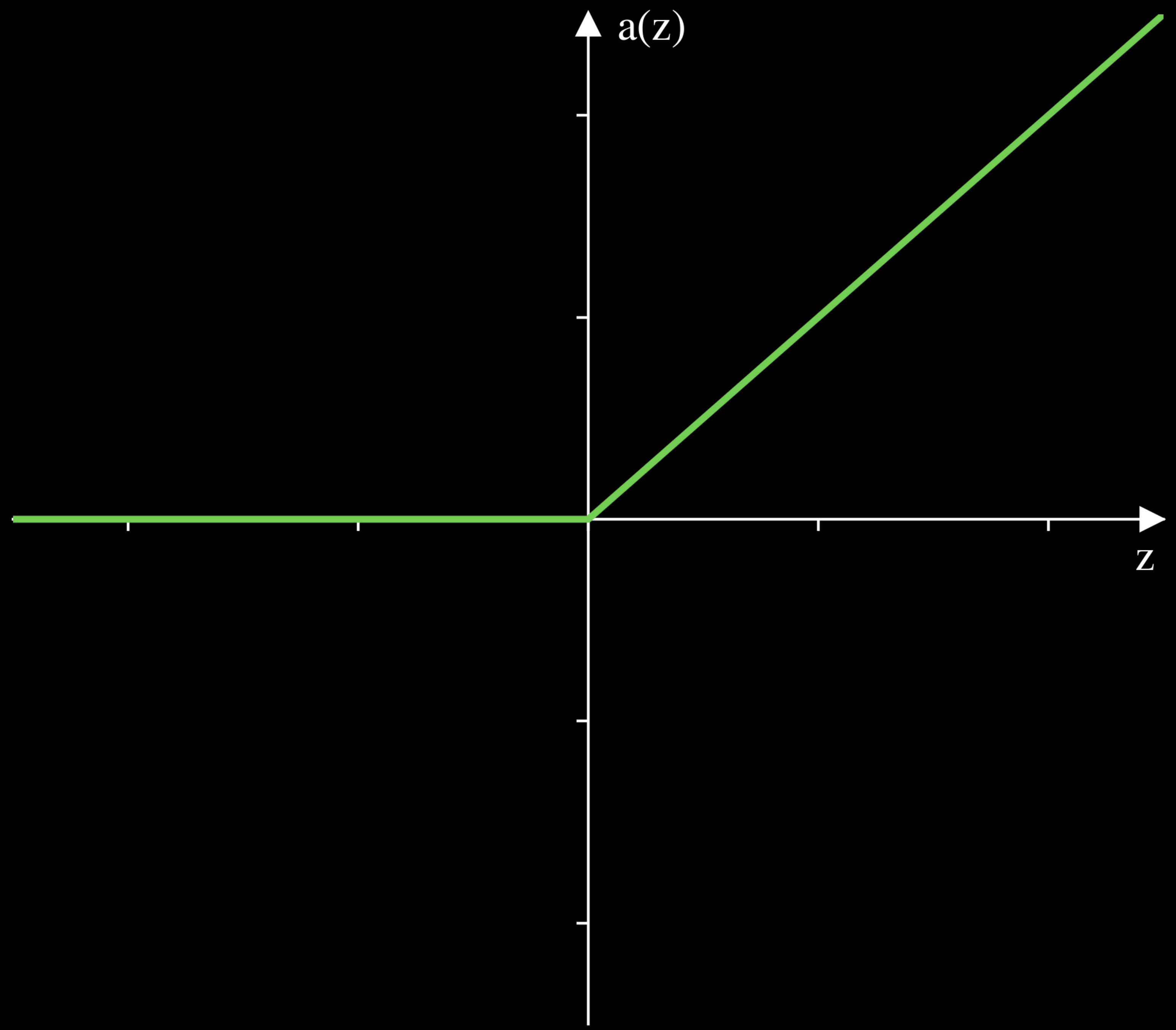
Autres fonctions d'activation

$a(z) = z$



Activation linéaire

$a(z) = max(0{,}z)$



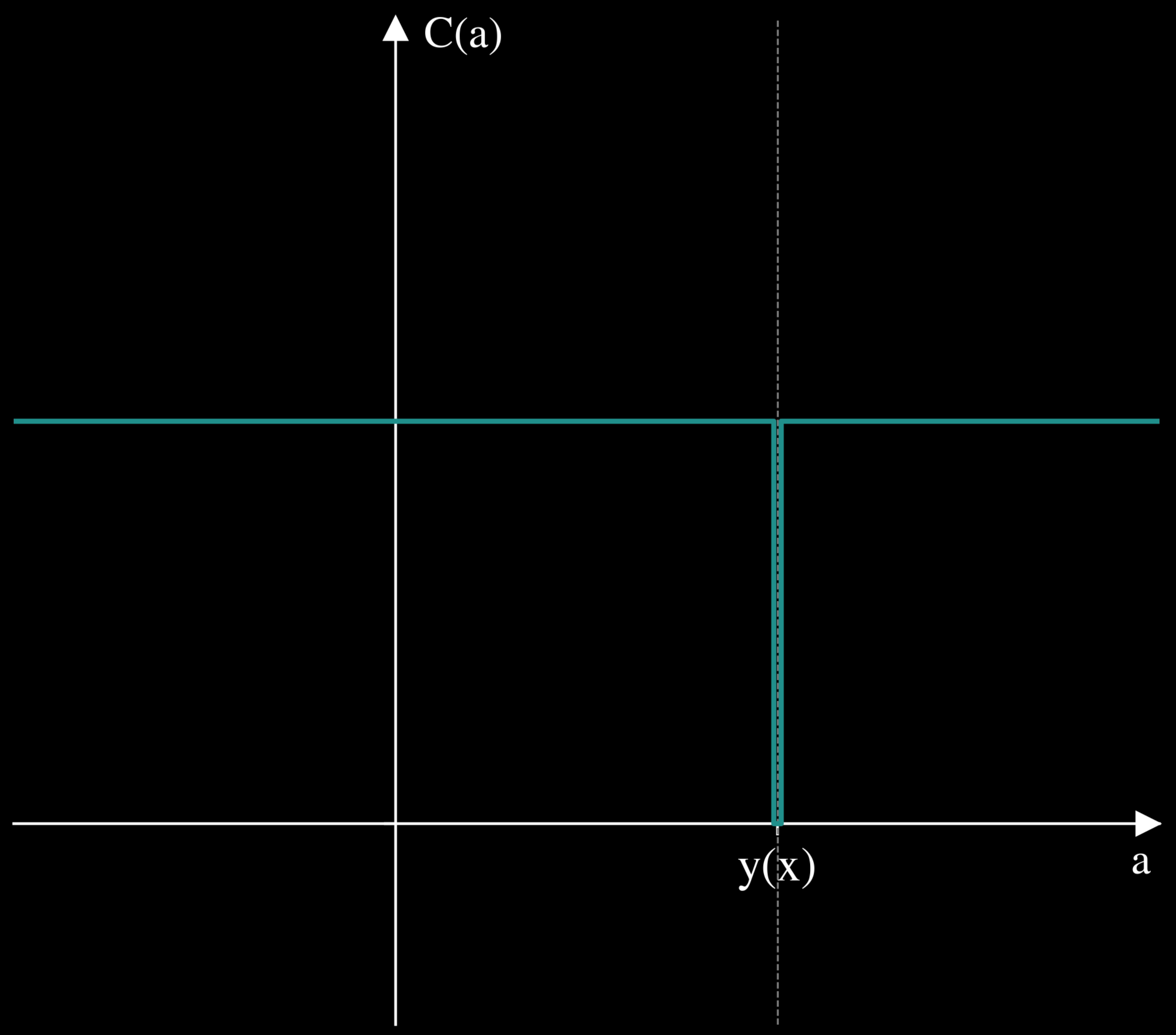
Activation ReLU

$a(z) = max(\varepsilon z{,}z)$

Activation Leaky ReLU

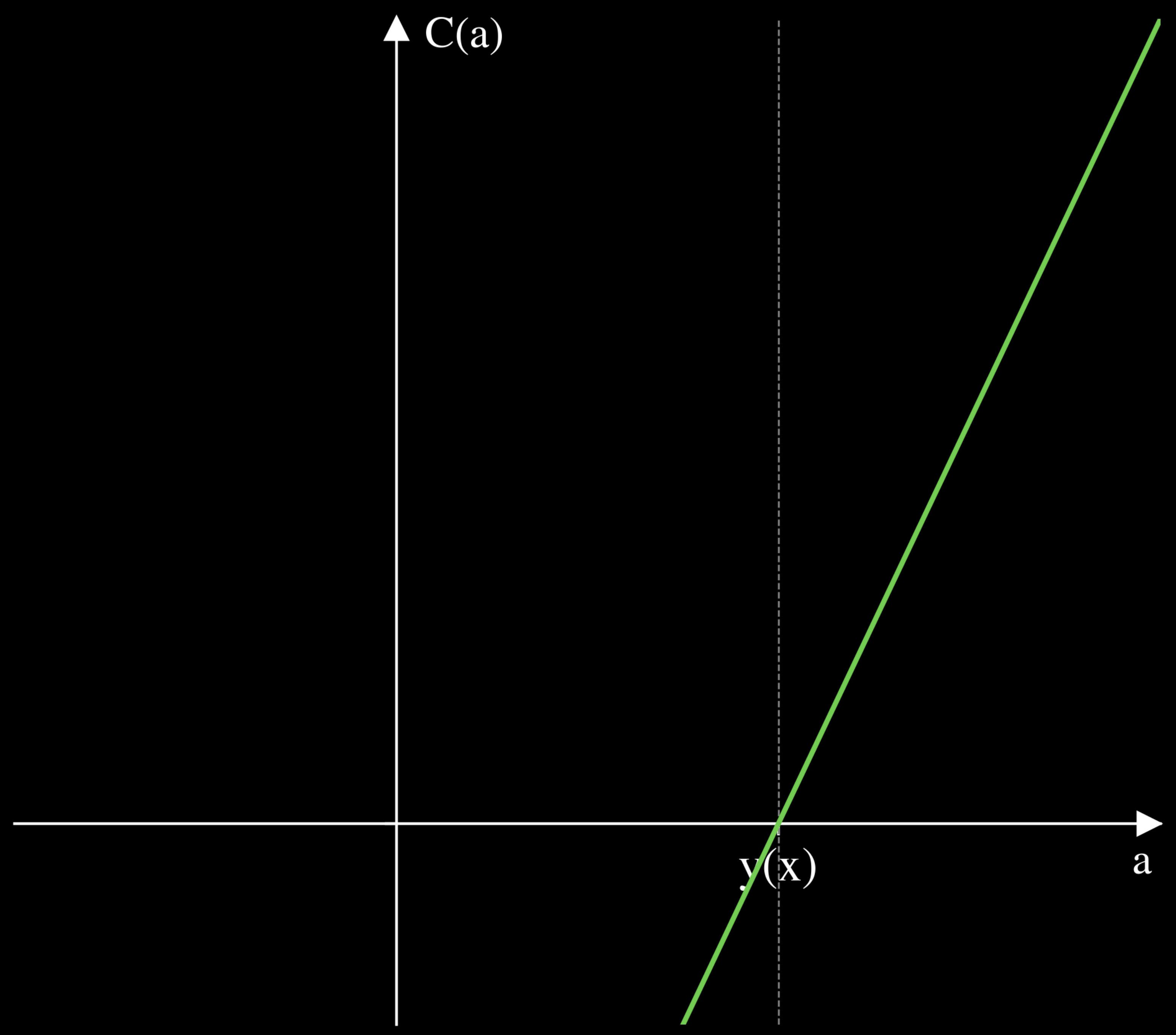
Autres fonctions de coût

$$C(w,b) = \begin{cases} 0 & \text{si } y(x) = a(w,b) \\ 1 & \text{si } y(x) \neq a(w,b) \end{cases}$$



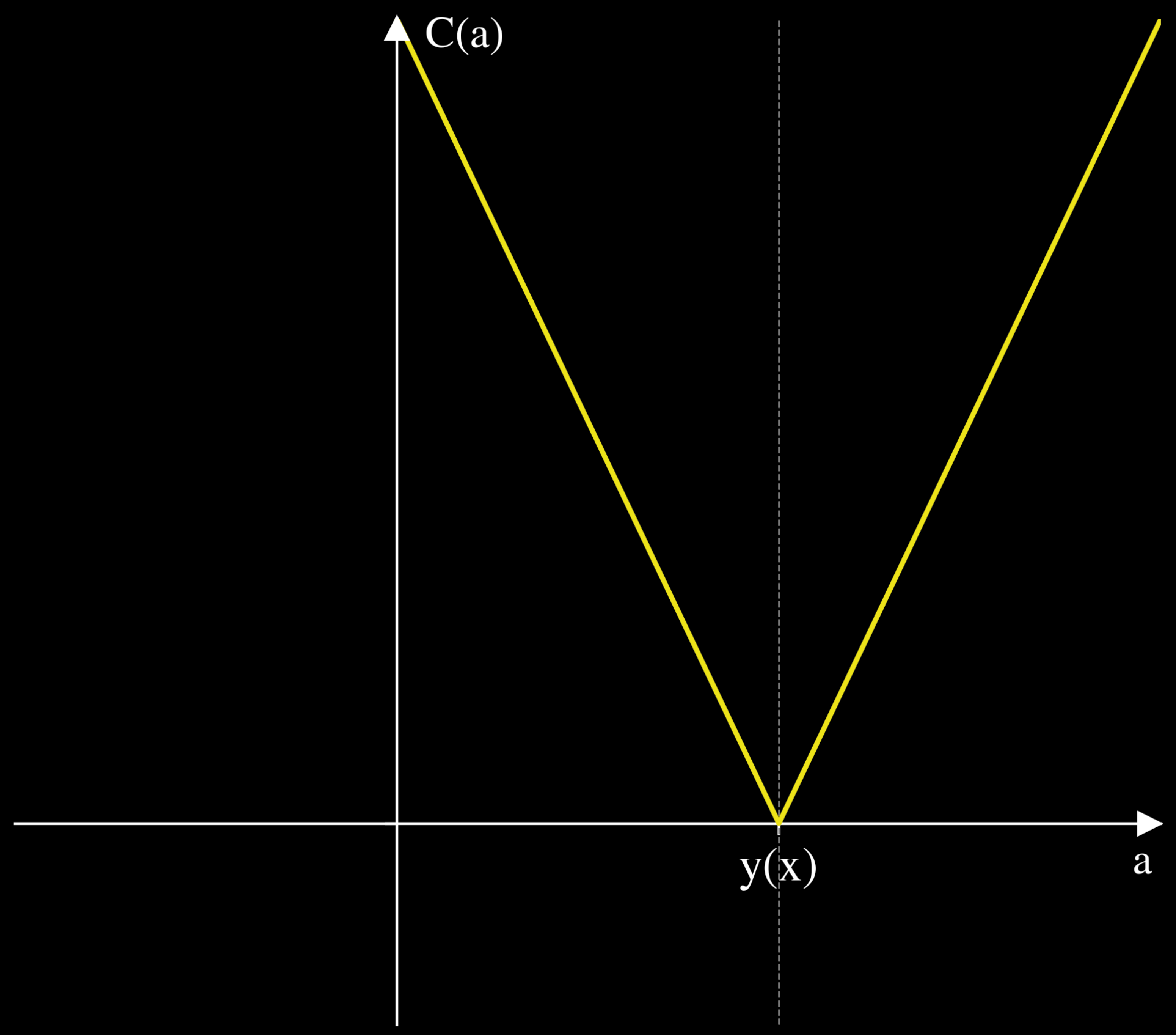
Erreur 0-1

$$C(w,b) = \frac{1}{n}\sum_x (y(x) - a(w,b))$$



Erreur moyenne

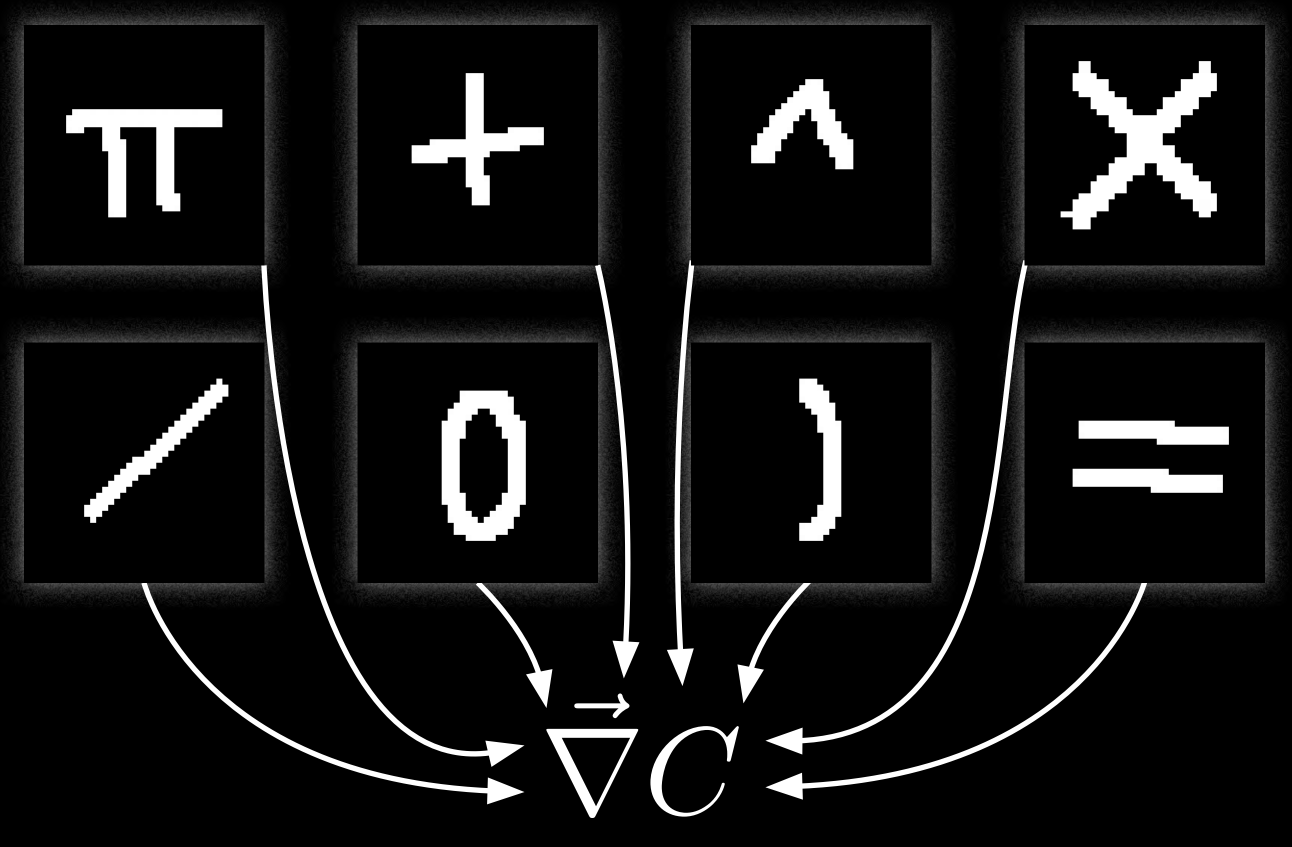
$$C(w,b) = \frac{1}{2n}\sum_x ||y(x) - a(w,b)||$$



Erreur moyenne absolue

Calcul du gradient basique :

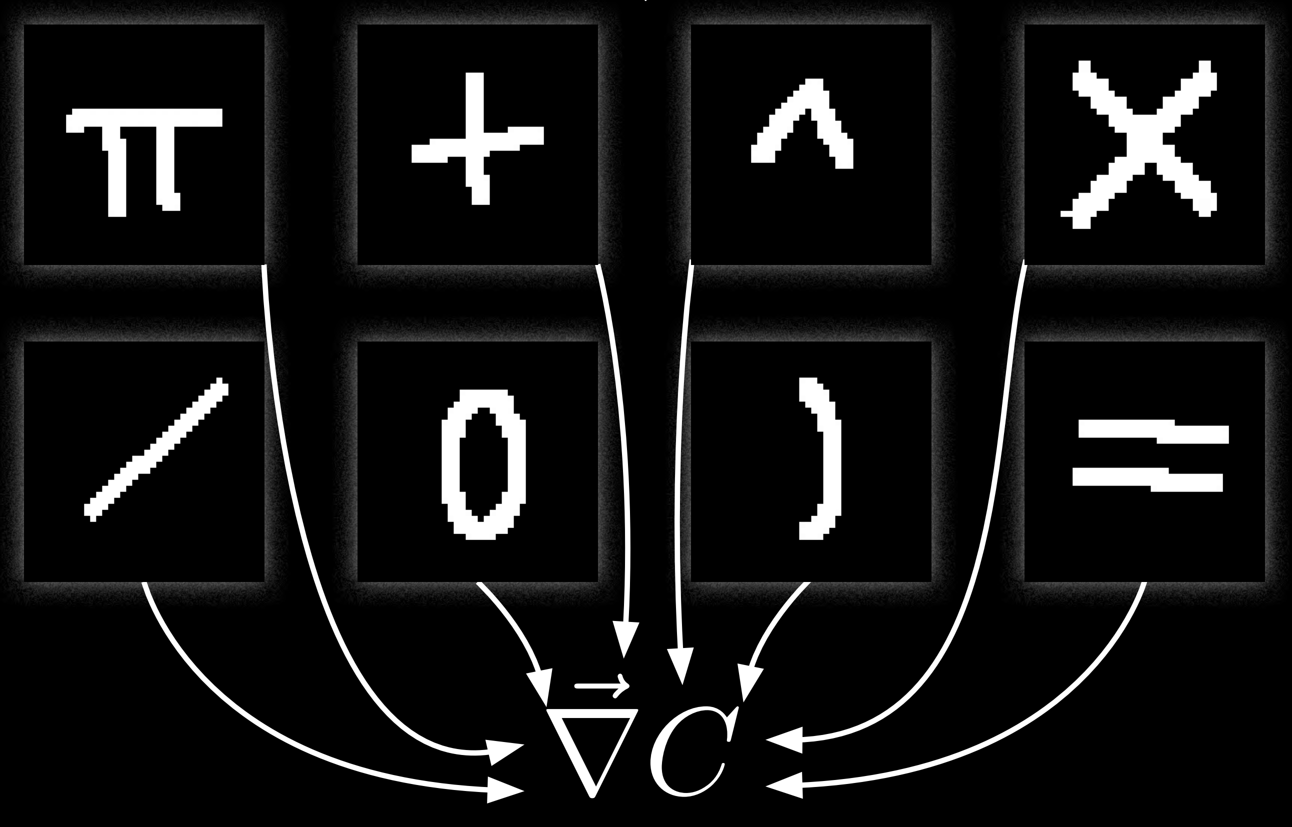
$$\vec{\nabla} C = \frac{1}{n} \sum_{x} \vec{\nabla} C_x \text{ avec n} \in [10^3; 10^6]$$



$\downarrow$

Changement des poids et biais du réseau à partir du gradient

$\downarrow$



$\downarrow$

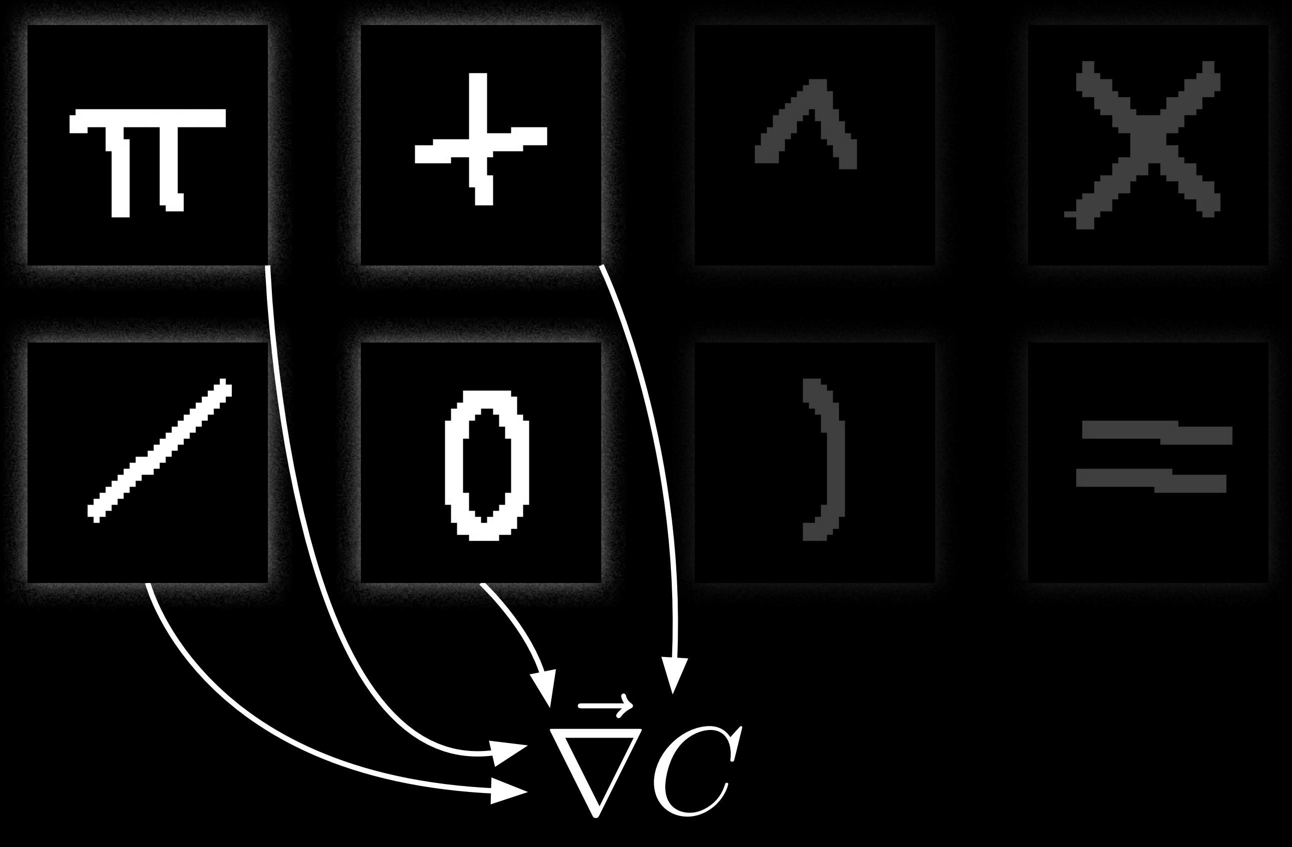
Changement des poids et biais du réseau à partir du gradient

$\downarrow$

...

Calcul du gradient stochastique :

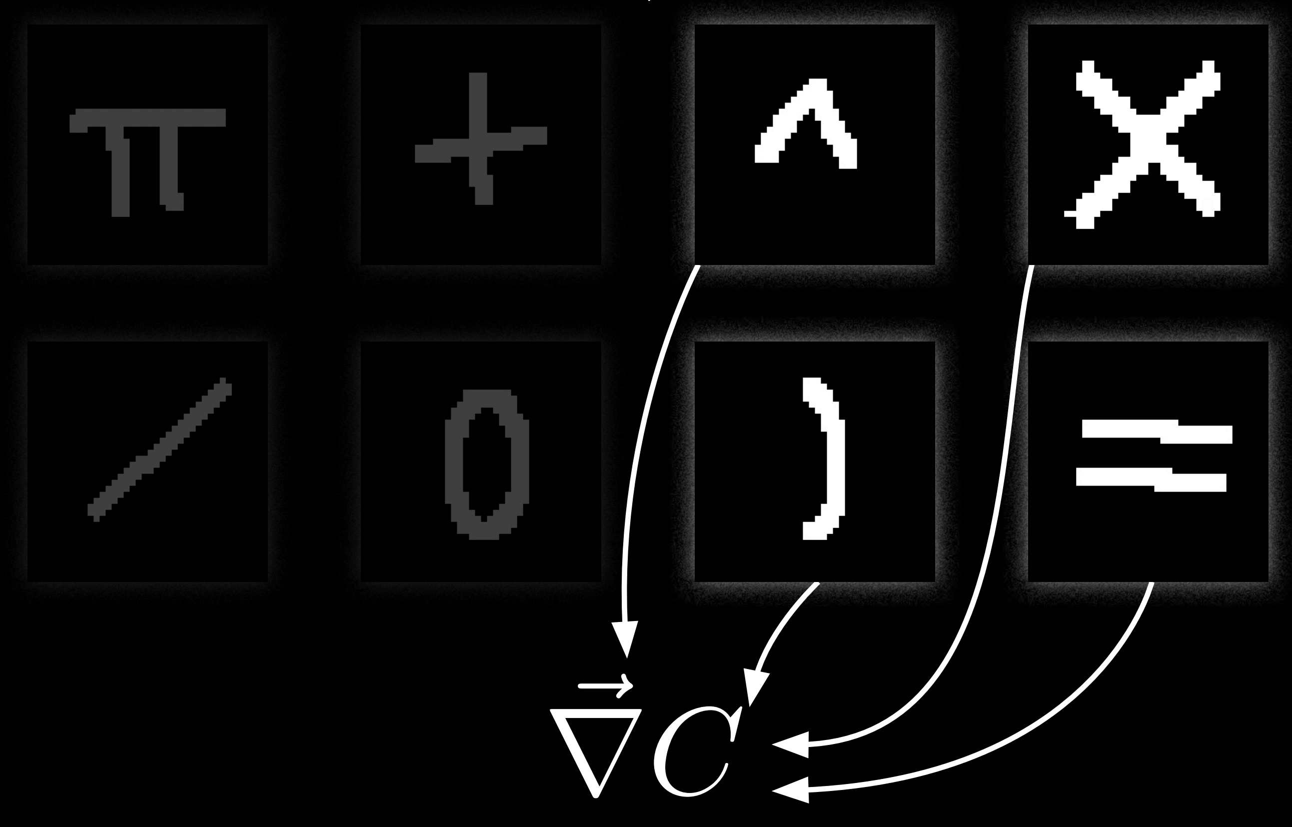
$$\vec{\nabla} C = \frac{1}{i} \sum_{x_i} \vec{\nabla} C_{x_i} \quad avec \quad \begin{cases} x_i \in x \\ i \simeq 32 \end{cases} \quad \ldots \quad \vec{\nabla} C = \frac{1}{j} \sum_{x_j} \vec{\nabla} C_{x_j} \quad avec \quad \begin{cases} x_j \in x \\ j \simeq 32 \end{cases}$$



$\downarrow$

Changement des poids et biais du réseau à partir du gradient

$\downarrow$

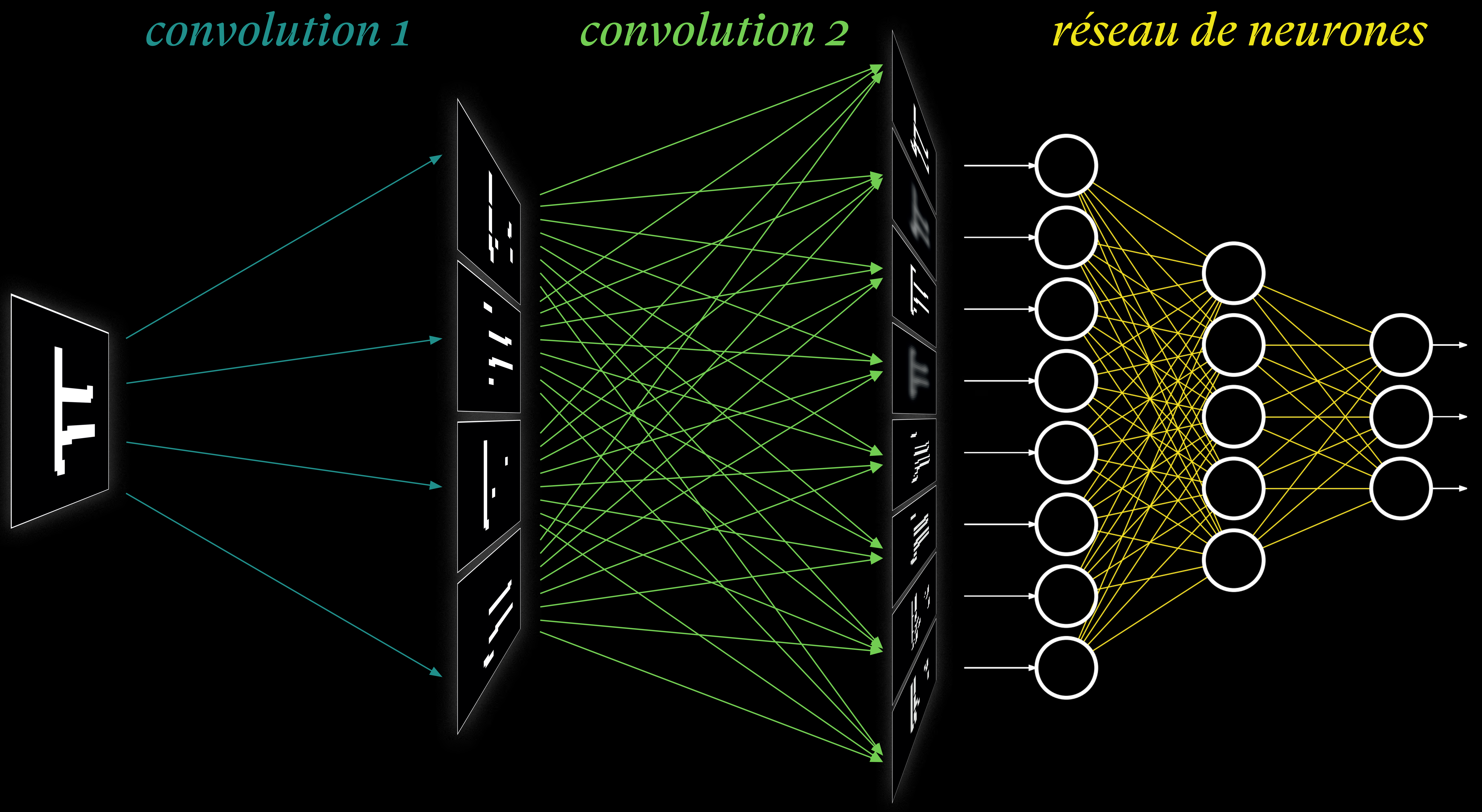


$\downarrow$

Changement des poids et biais du réseau à partir du gradient

$\downarrow$

...


convolution 1
convolution 2
réseau de neurones

Bibliographie :

- *I. Goodfellow, Y. Bengio, A. Courtville :* **Deep Learning**
- *M. Nielsen :* **Neural Networks & Deep Learning**
- *3Blue1Brown :* **Neural Networks**
- *C. Azencott :* **Introduction au Machine Learning**
- *A. Cornuéjols, L. Miclet:* **Apprentissage artificiel : concepts & algorithmes**