

Génération d'images par apprentissage profond

GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Génération d'images par apprentissage profond

Bref historique de l'intelligence artificielle (IA)

GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Qu'est-ce que l'intelligence artificielle?

« L'ensemble de théories et de techniques
mises en œuvre en vue de réaliser
des *machines capables de simuler l'intelligence*. »

(Larousse)

Bref historique...

Naissance de l'intelligence artificielle



1956

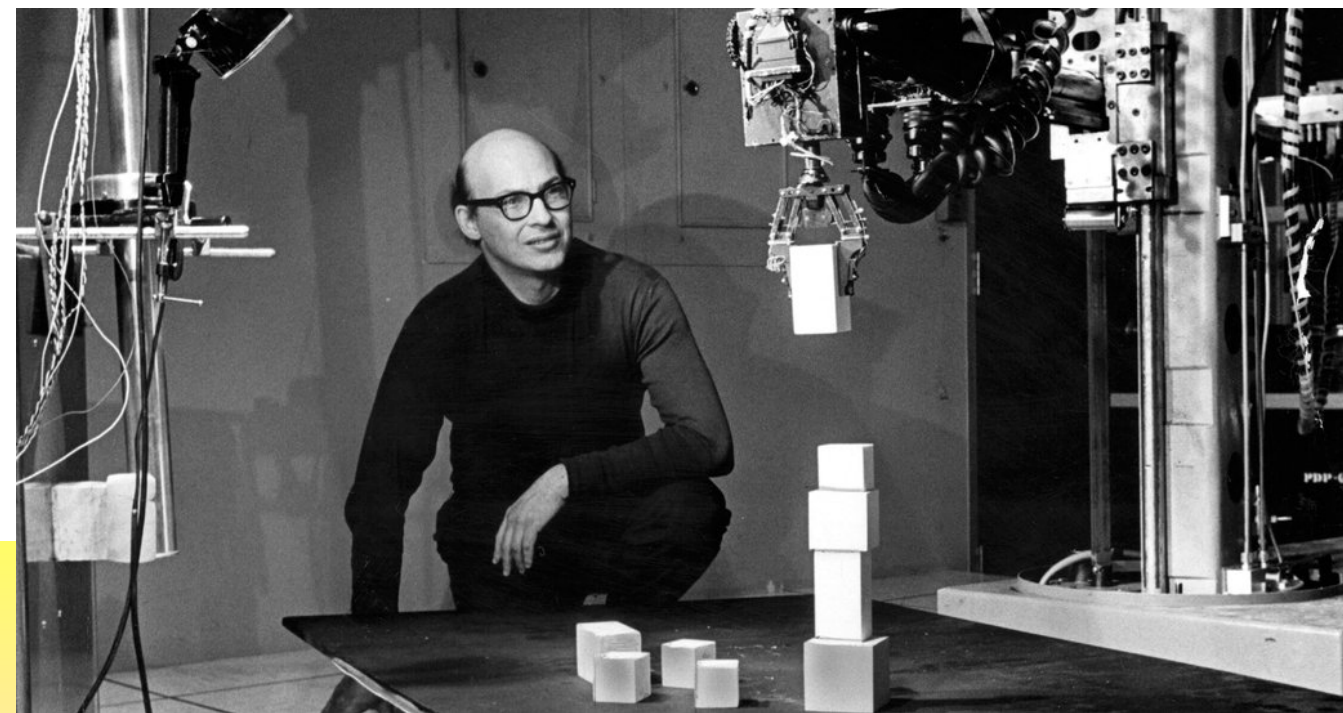
Bref historique...

e de l'intelligence artificielle

Âge d'or

Premiers réseaux de neurones
Logique formelle
Compréhension des langues
« blocks world »

Nous aurons une
machine d'intelligence
humaine d'ici 3 à 8 ans!

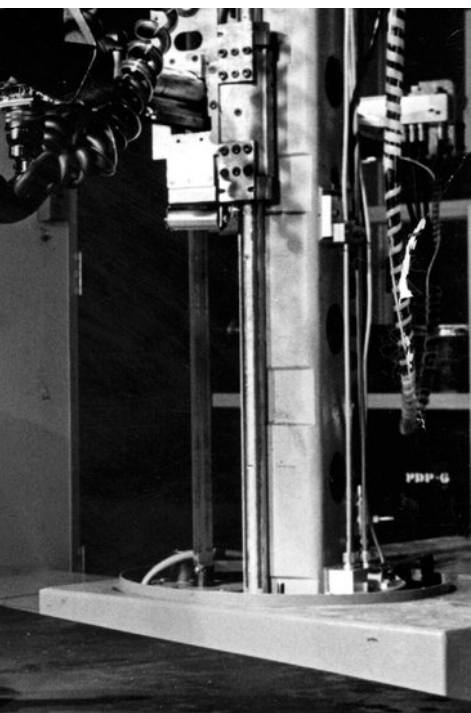


1956

1974

Bref historique...

r
e neurones
nelle
es langues
rld »



Nous aurons une
machine d'intelligence
humaine d'ici 3 à 8 ans!

Hiver de l'IA

Promesses
trop ambitieuses!

Ordinateurs
pas performants

1974

1980

1987

Bref historique...

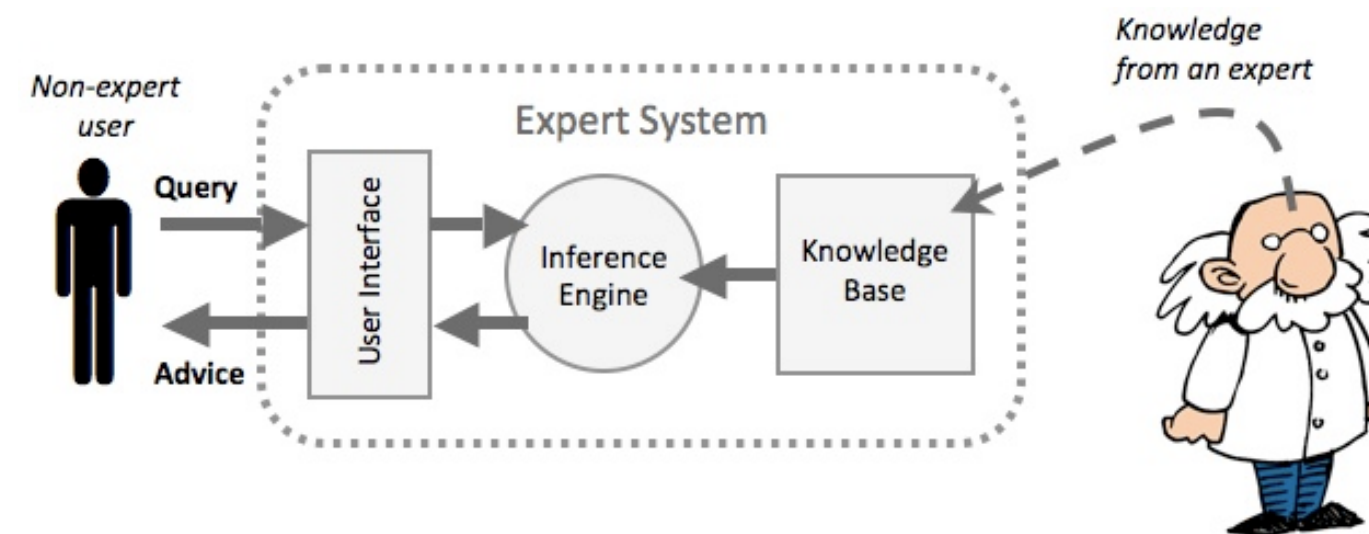
Hiver de l'IA

Promesses
trop ambitieuses!

Ordinateurs
pas performants

Regain d'intérêt

Systèmes experts
Réseaux de neurones
refont surface



1974

1980

1987

1993

Bref historique...

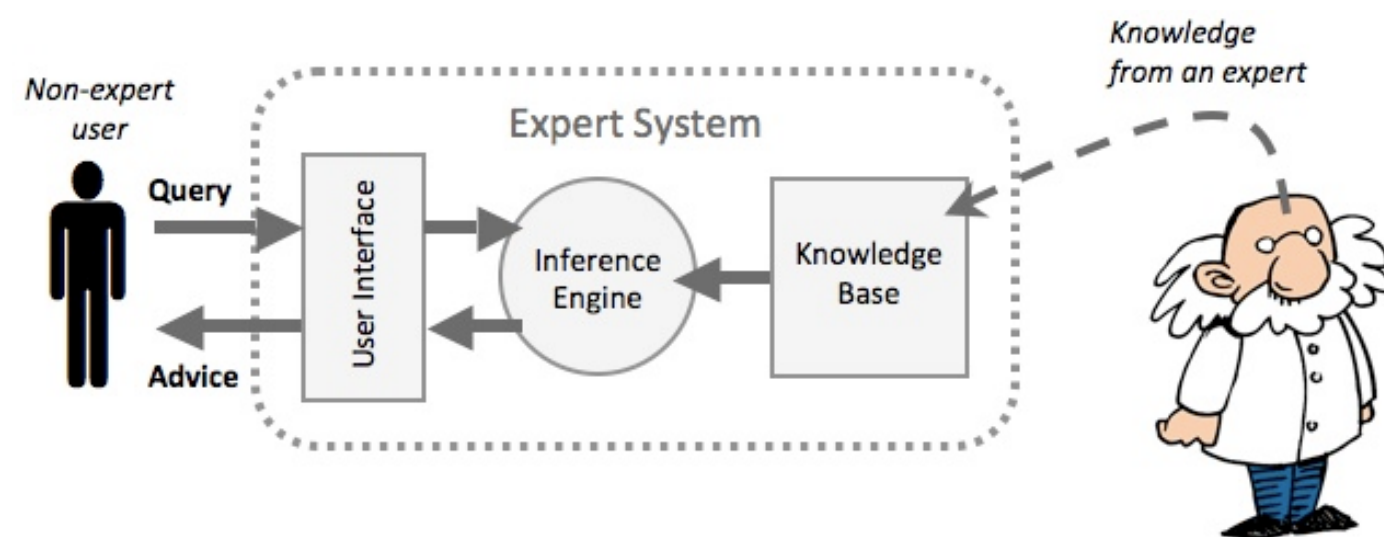
l'IA

ses
euses!

urs
nants

Regain d'intérêt

Systemes experts
Réseaux de neurones
refont surface



2e hiver de l'IA

Systemes experts
pas assez robustes

Réseaux de neurones
battus par d'autres
techniques

1980

1987

1993

Bref historique...

2e hiver de l'IA

Systèmes experts
pas assez robustes

Réseaux de neurones
battus par d'autres
techniques

Les promesses renouvelées

Ordinateurs plus performants
(10 millions de fois plus qu'en 1950!)

IBM Deep Blue



Premières voitures autonomes



1993

2010

Bref historique...

Les promesses renouvelées

ordinateurs plus performants
100 fois plus qu'en 1950!

Premières voitures autonomes

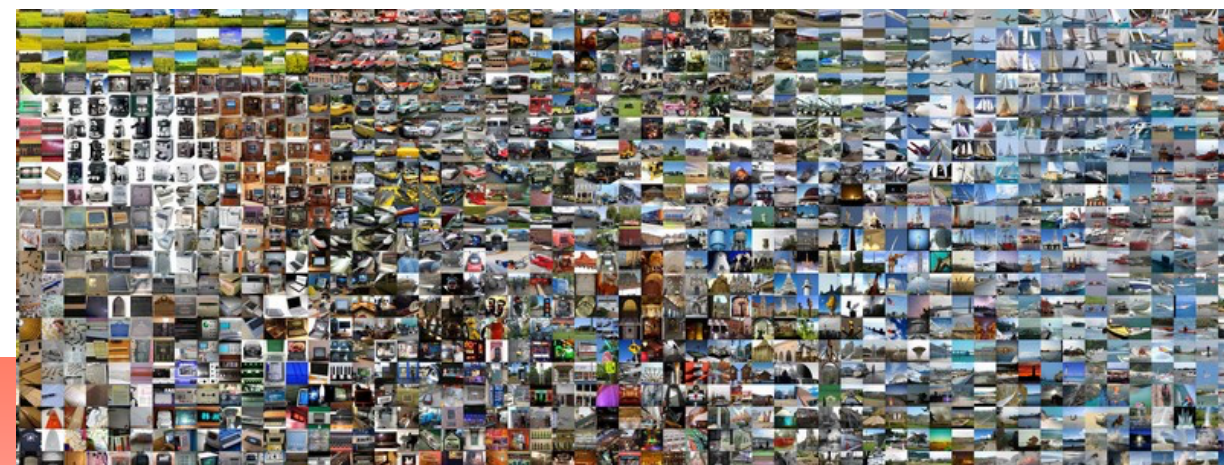


L'explosion...

Réseaux de neurones profonds
(et algorithmes efficaces)

Capacités de calcul (GPU)

Base de données massives



2010

Génération d'images par apprentissage profond

Introduction à l'apprentissage profond

GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Quelques statistiques

Rappel



2014

250 milliard d'images au total
350 millions téléversées par mois



2014

1.8 milliards d'images téléversées chaque jour



2021

400 heures téléversées à chaque minute



2015

2 milliard d'images

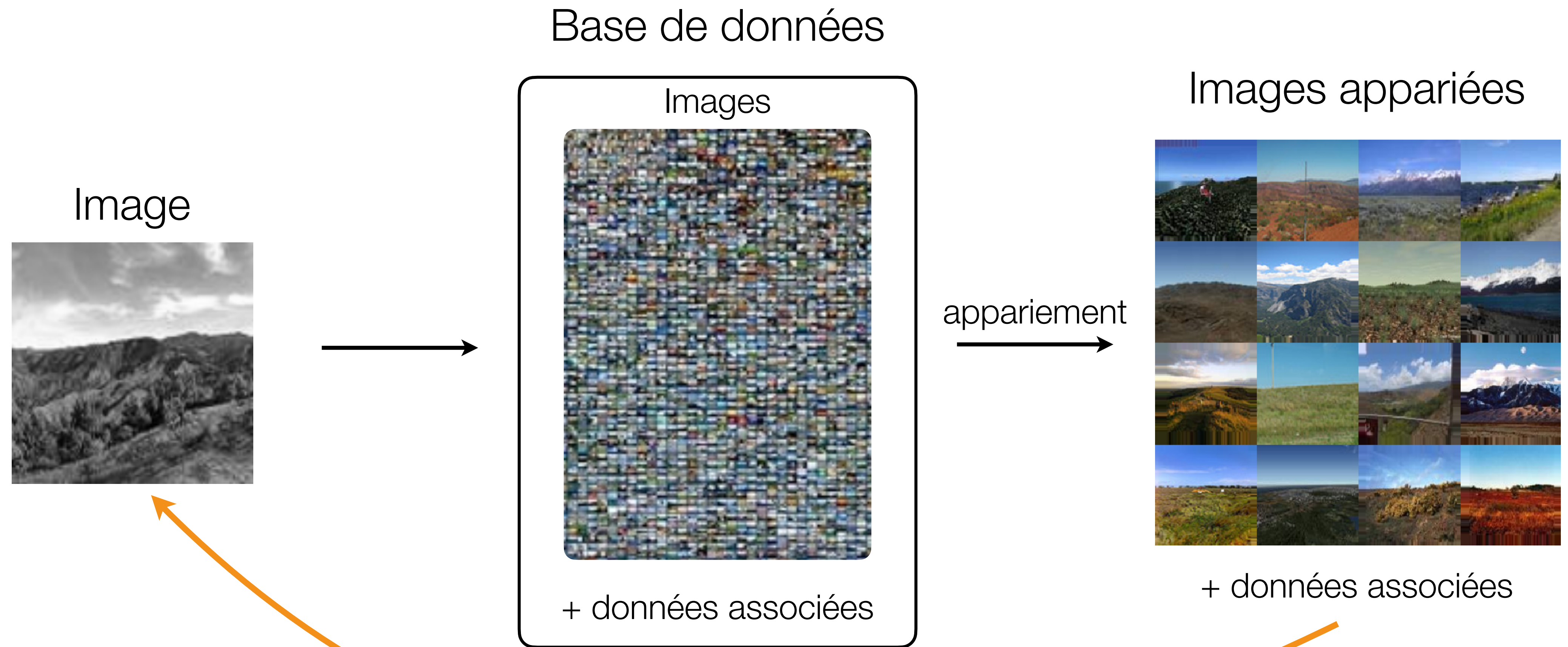


6 milliard d'images

Et aujourd'hui?

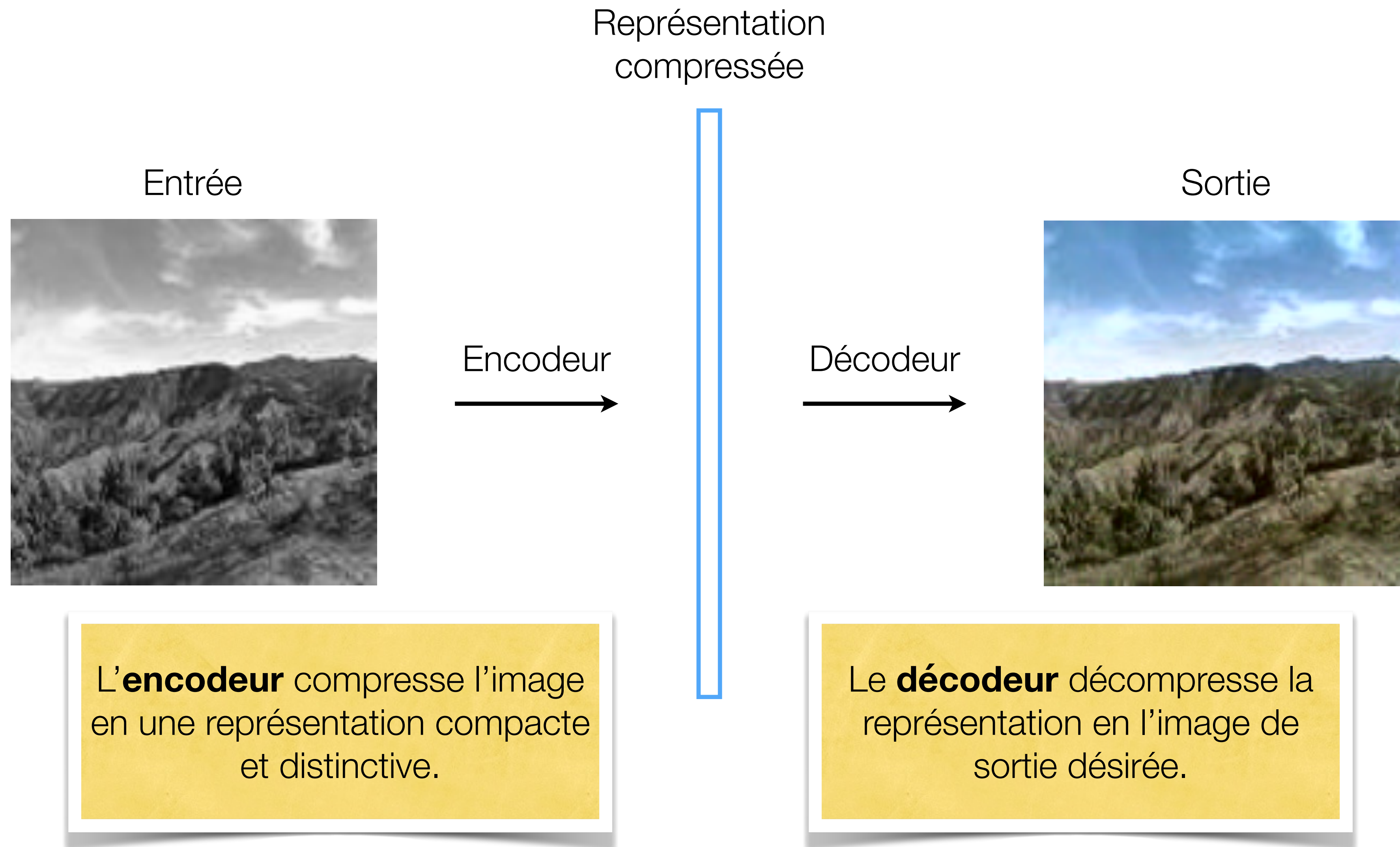
On ne compte plus...

Comment utiliser ces données?

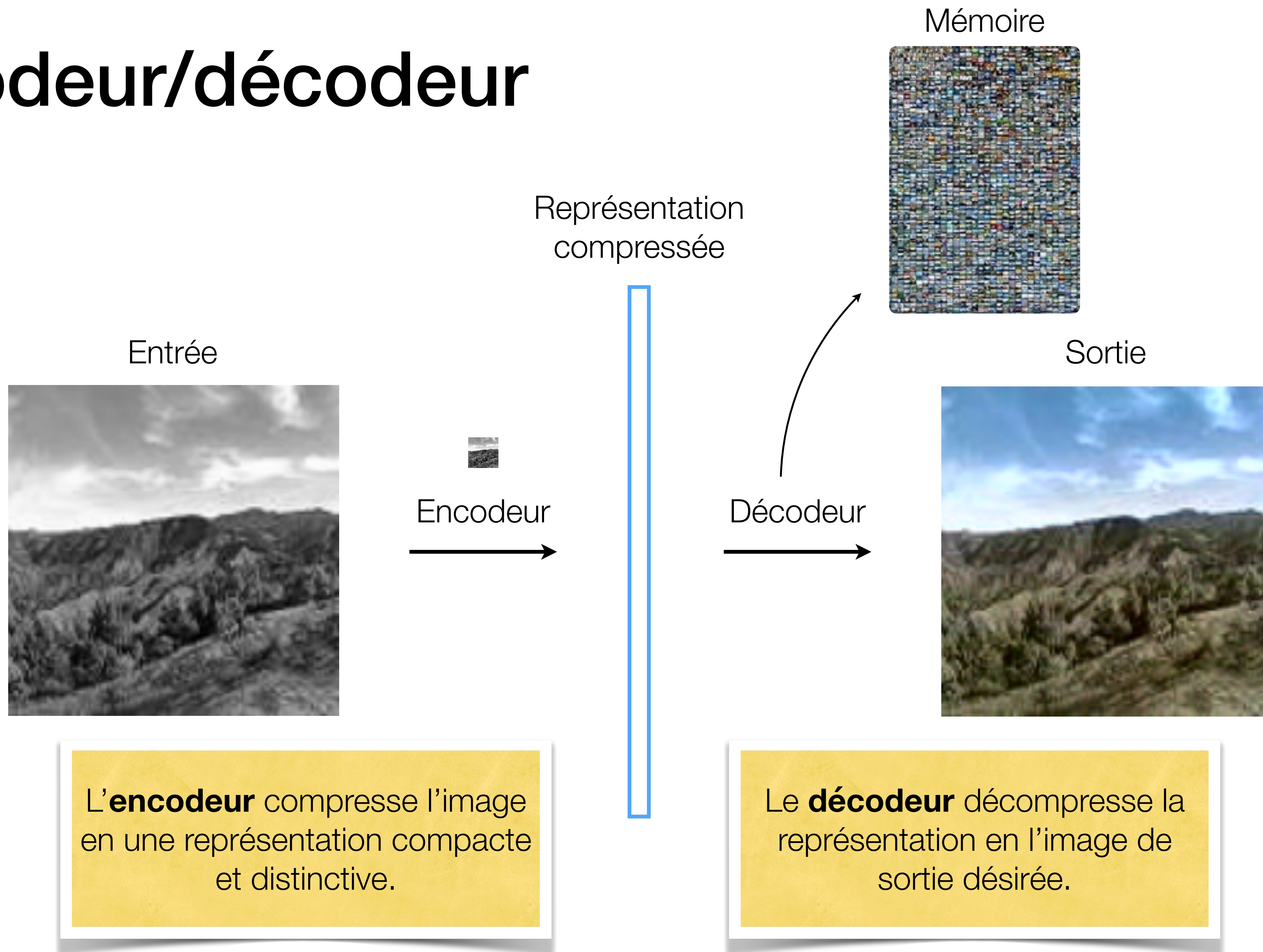


Si base de données est assez grande,
elle devrait contenir des images similaires!

Autre point de vue : encodeur/décodeur

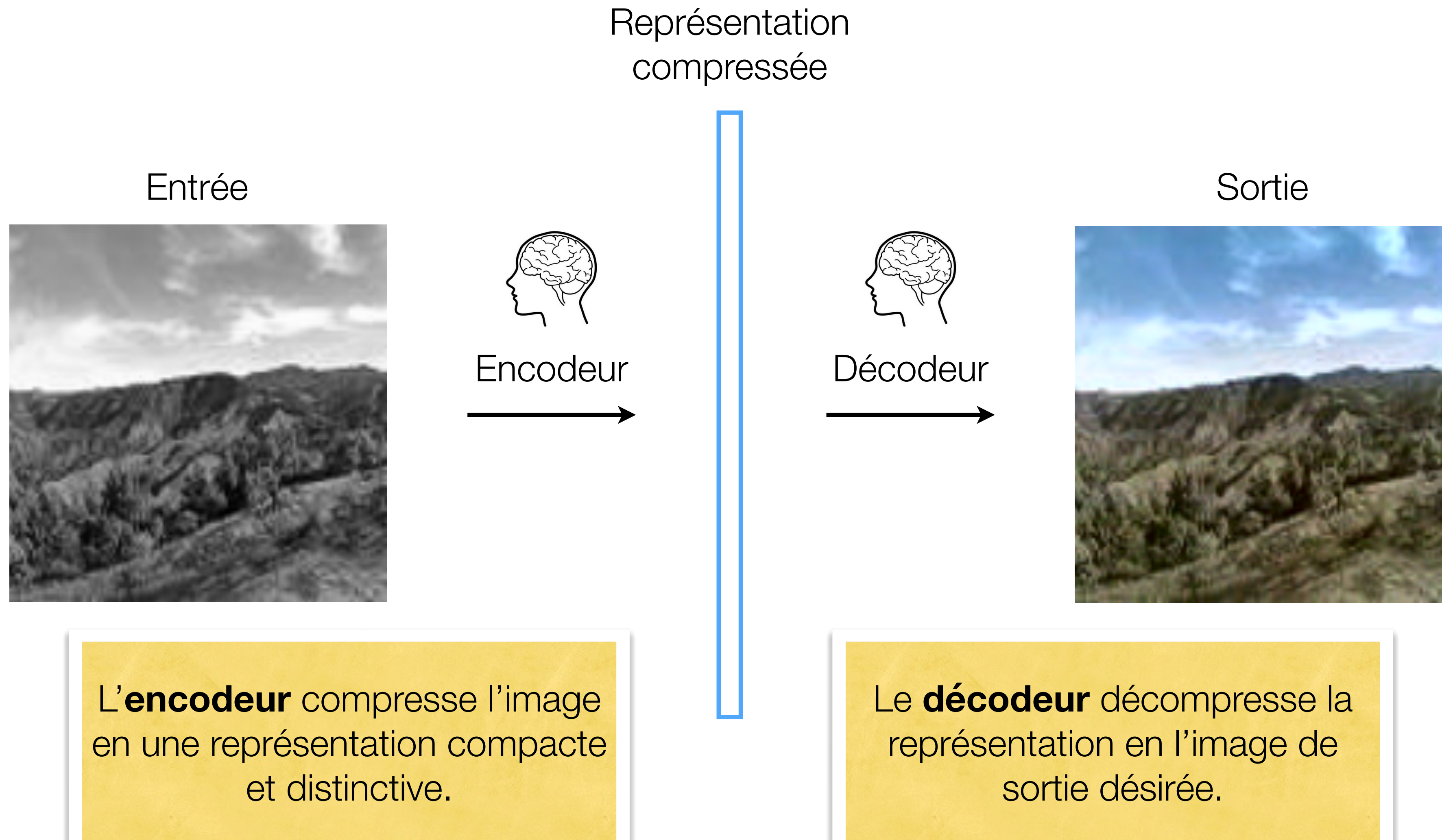


Encodeur/décodeur



Encodeur/décodeur

Apprendre l'encodeur et le décodeur
(à partir des données)
afin de satisfaire un objectif, ici : colorer l'image.





Stocke toute l'information
dans une mémoire

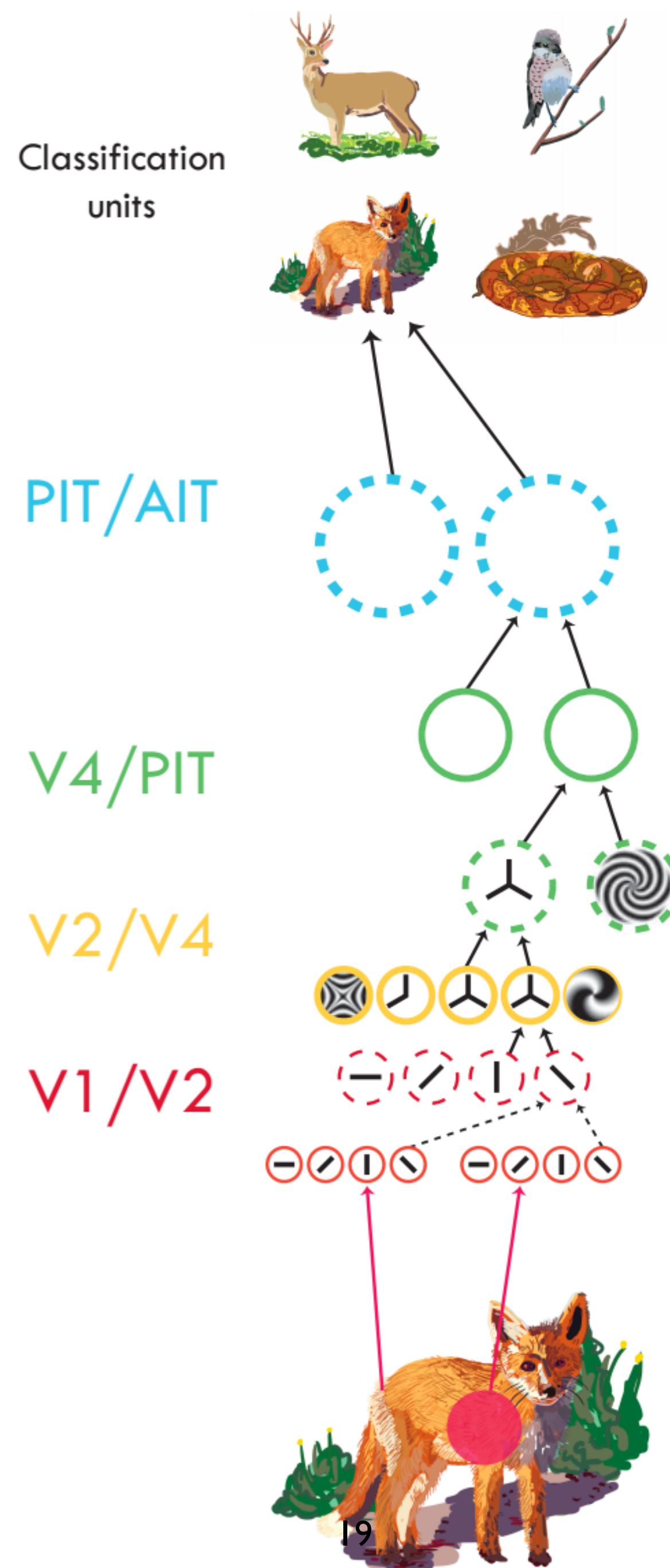
Représentation simple
permettant d'apparier les images



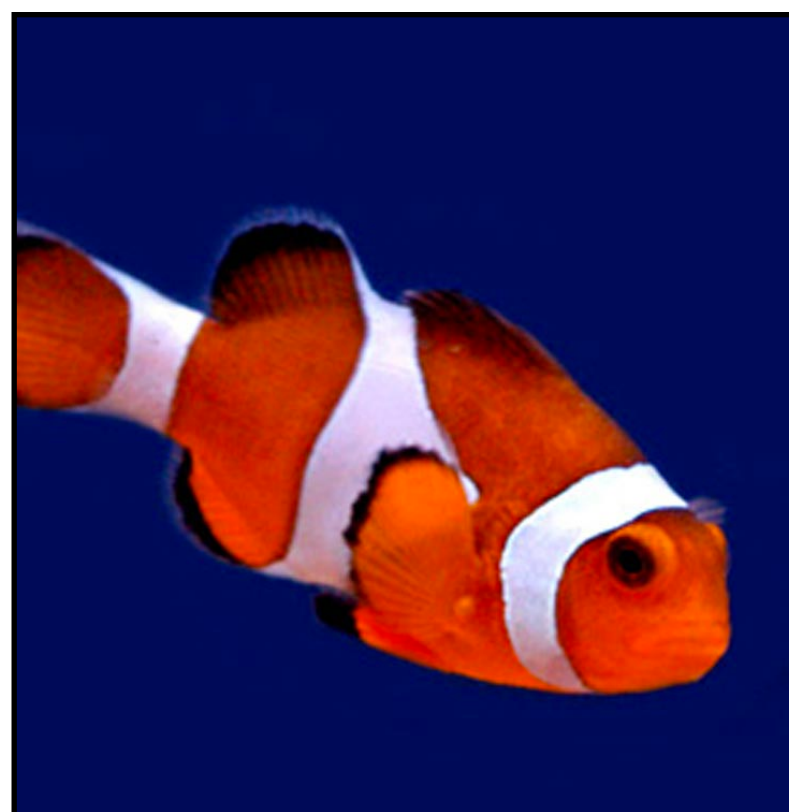
Assume que le monde
possède une structure simple

Apprend une représentation
qui capture cette structure

Structure **hiérarchique**
composées d'unités de calcul **simples**



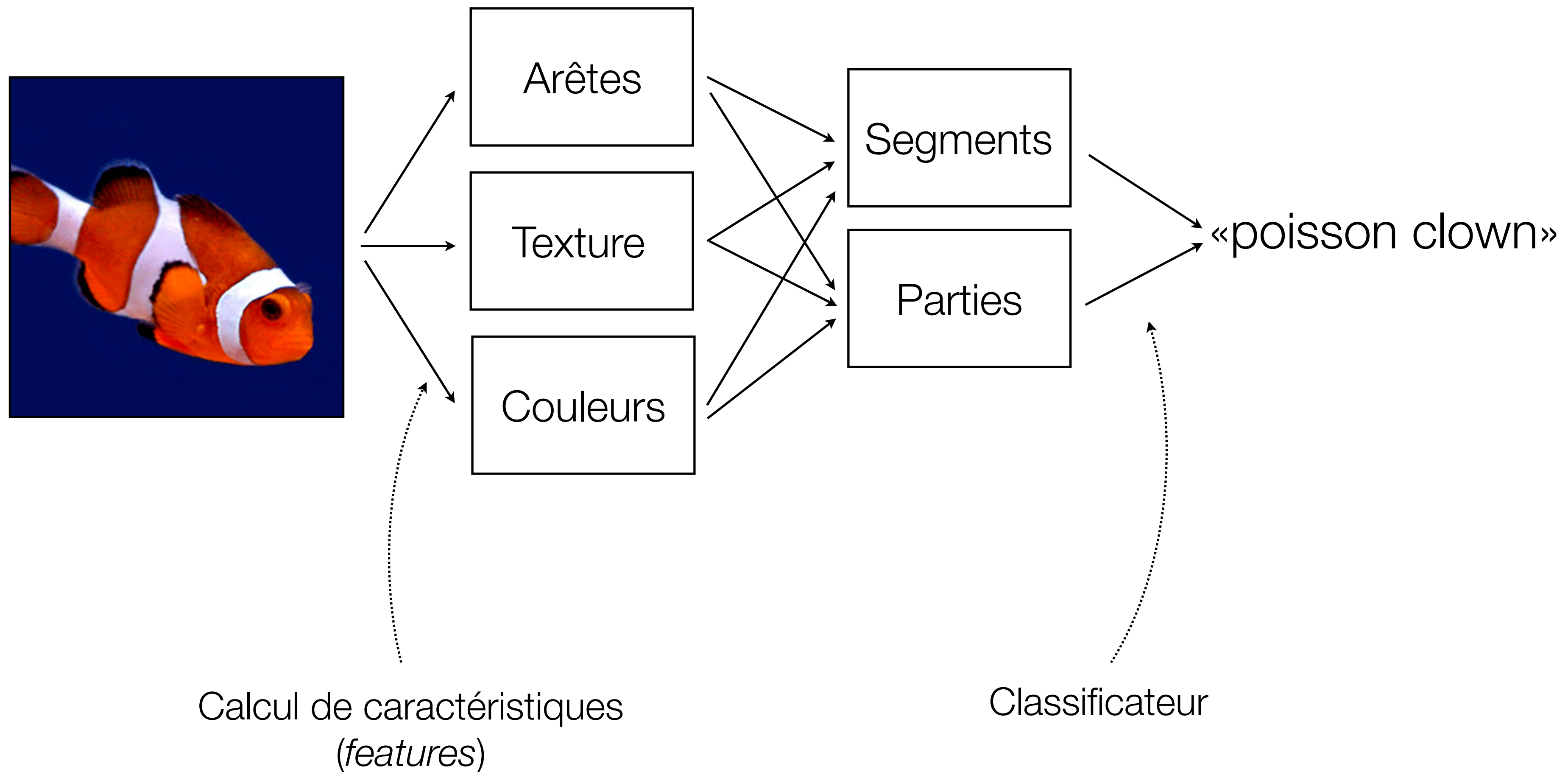
Reconnaissance d'images



«poisson clown»

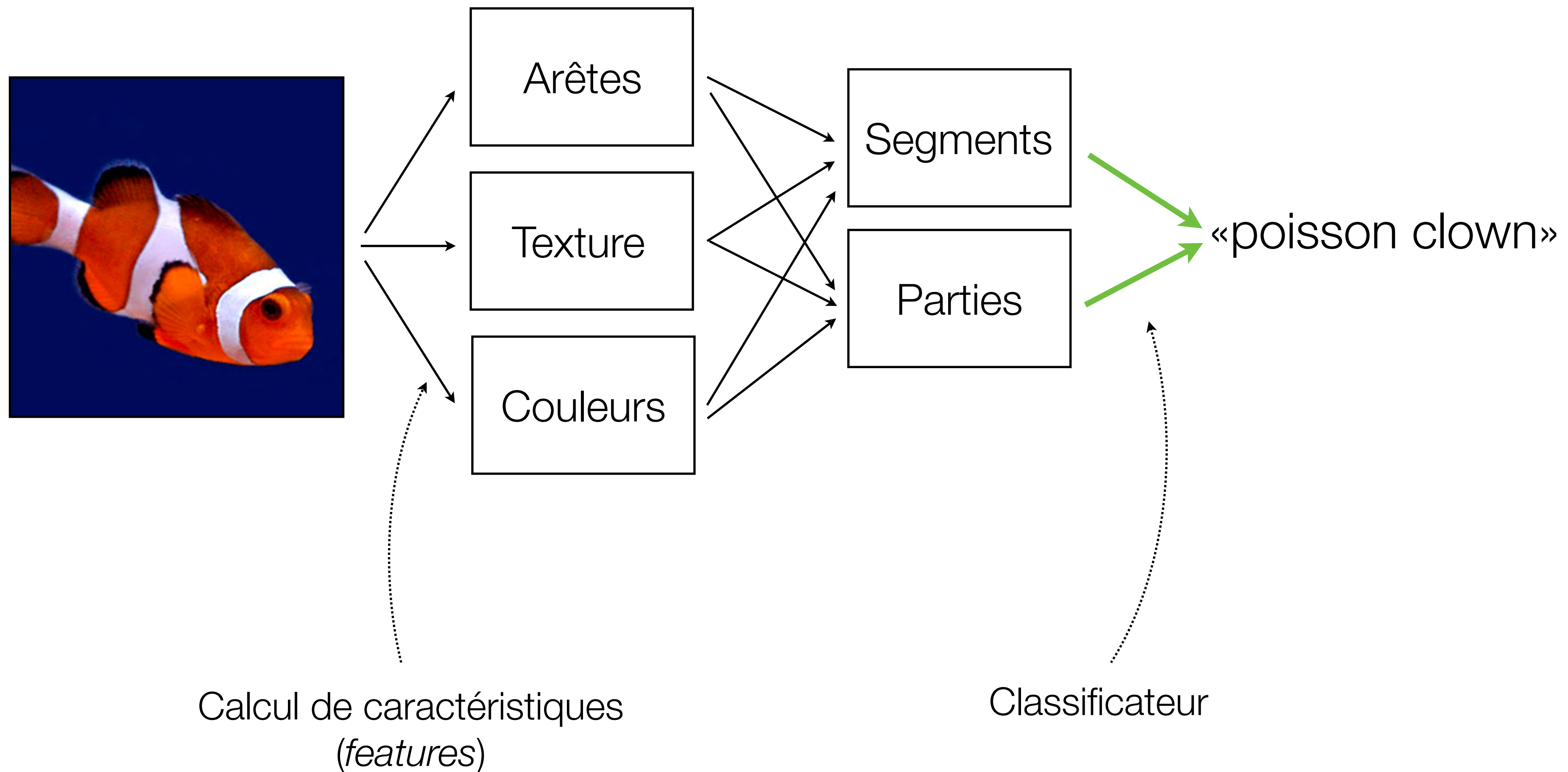
Structure **hiérarchique**
composées d'unités de calcul **simples**

Reconnaissance d'images : approche «traditionnelle»



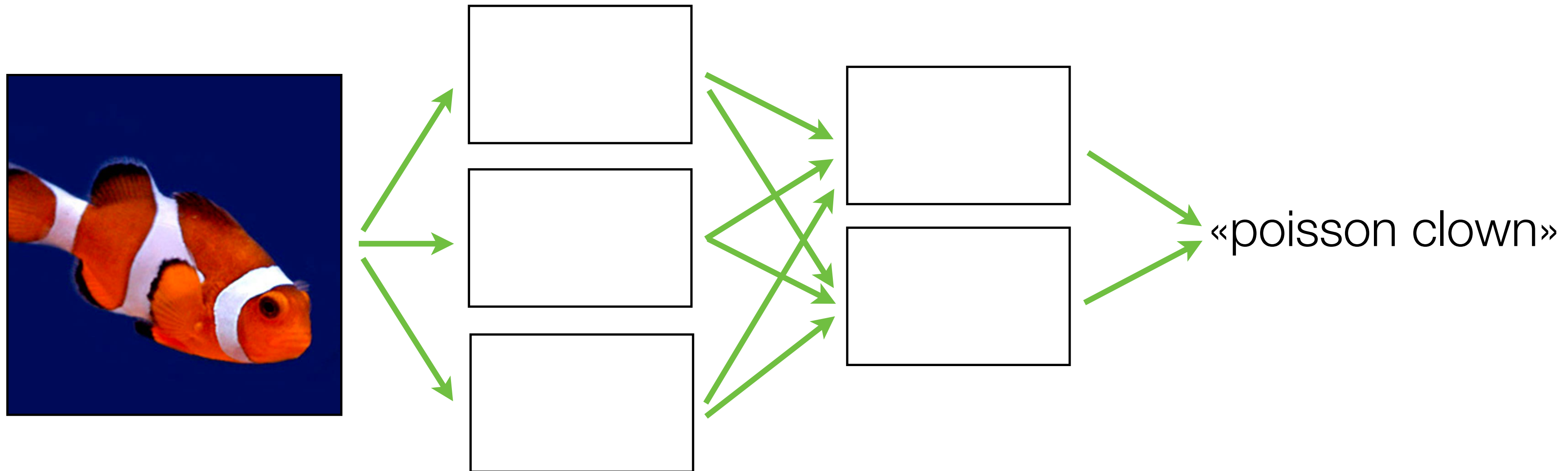
Reconnaissance d'images : approche «traditionnelle»

Appris automatiquement



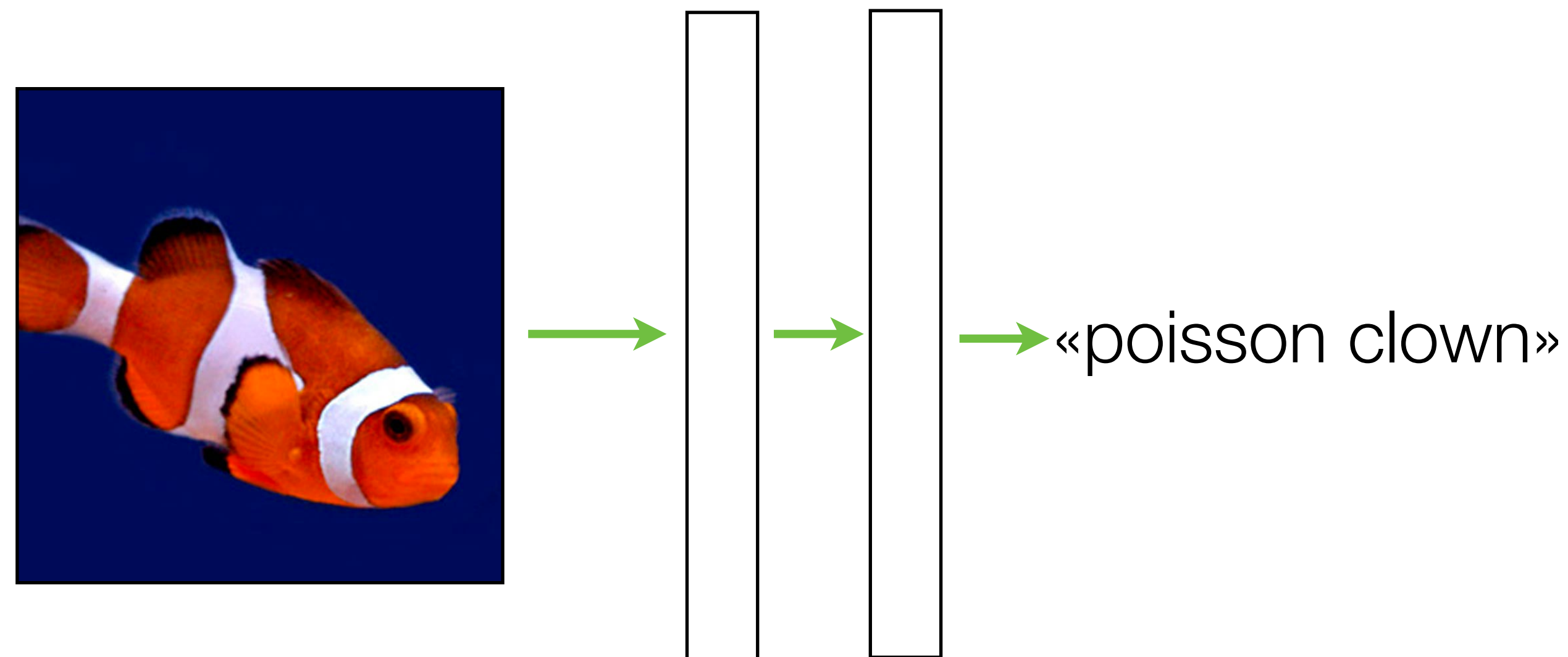
Réseau de neurones

Appris automatiquement



Réseau de neurones

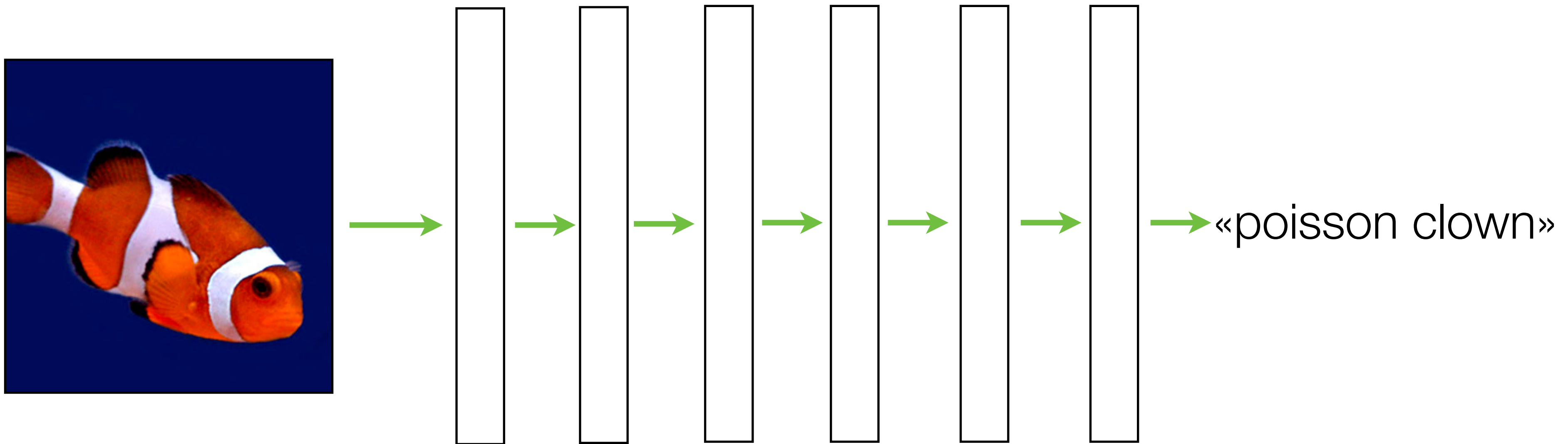
Appris automatiquement



Structure **hiérarchique**
composées d'unités de calcul **simples**

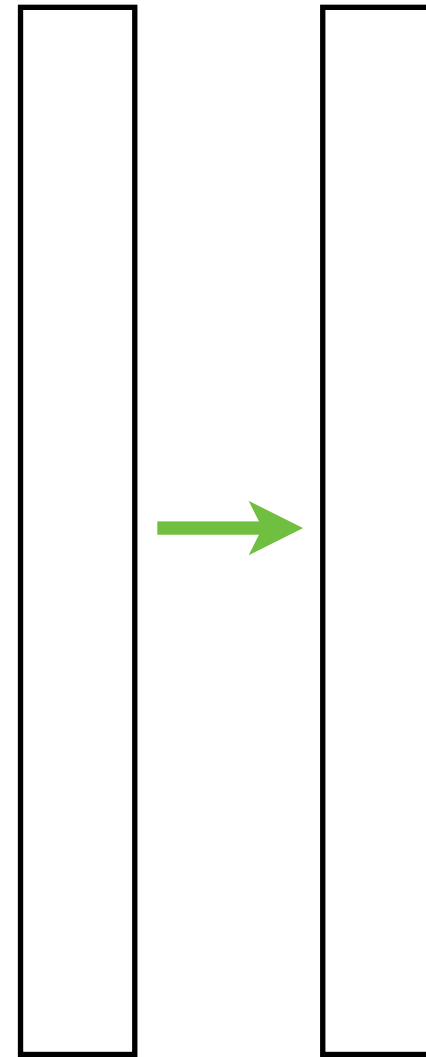
Réseau de neurones profond

Appris automatiquement



Structure **hiérarchique**
composées d'unités de calcul **simples**

Réseau de neurones *profond*



Calculs dans un réseau de neurones

Que font ces flèches?
Quels calculs sont effectués?



Génération d'images par apprentissage profond

Le perceptron

GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Calculs dans un réseau de neurones

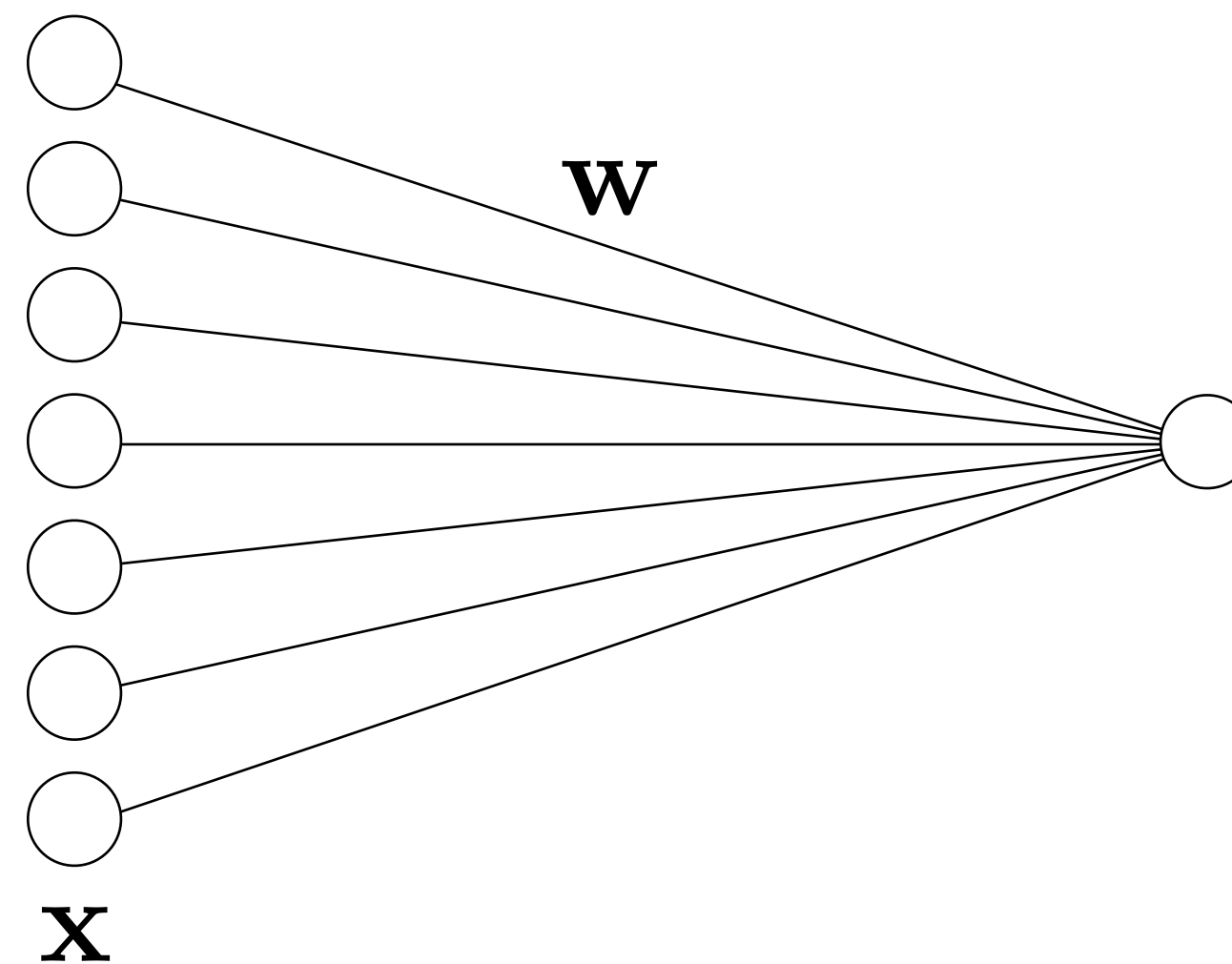
Que font ces flèches?
Quels calculs sont effectués?



Calculs dans un perceptron

Entrée

Sortie

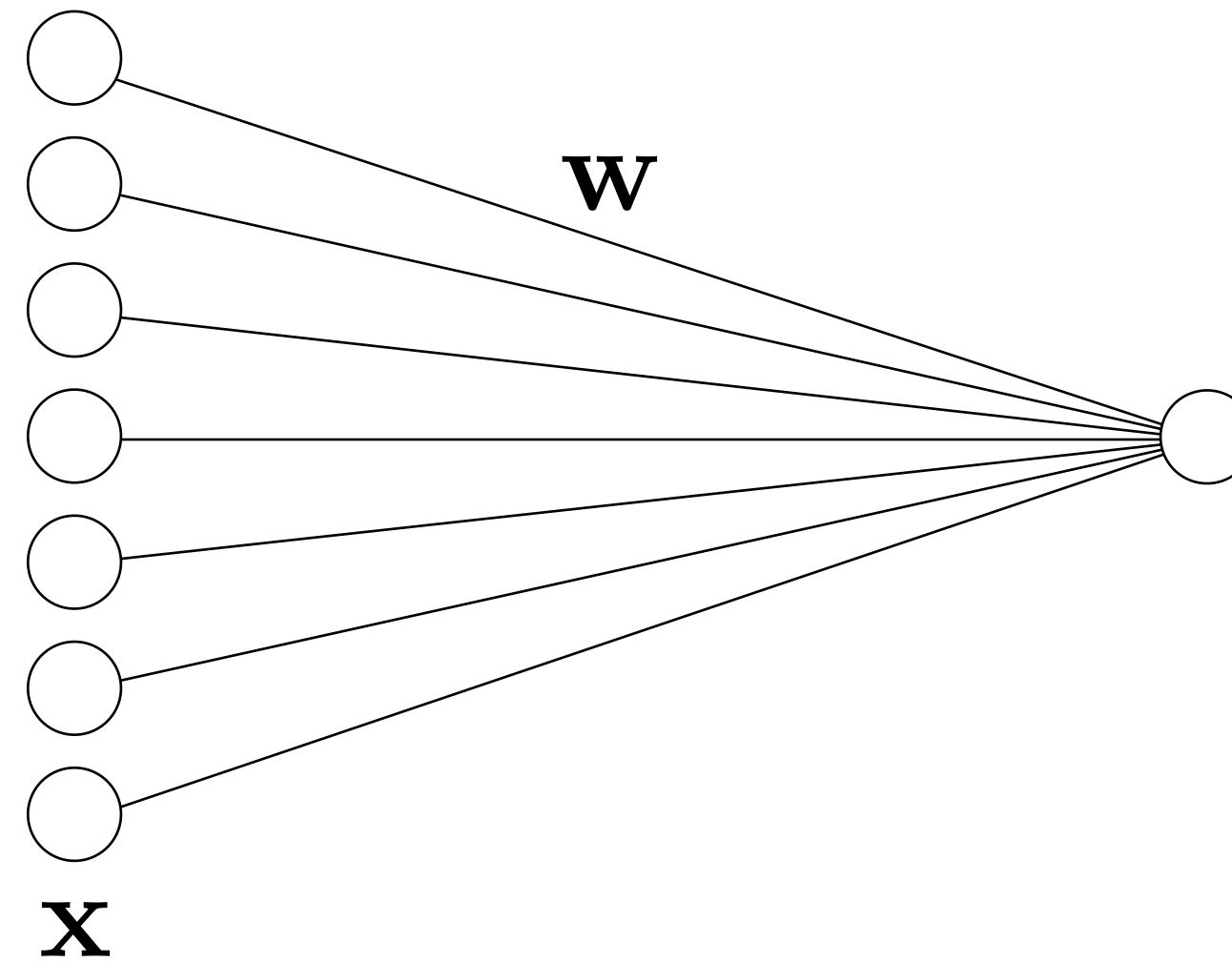


$$\sum_i w_i x_i + b$$

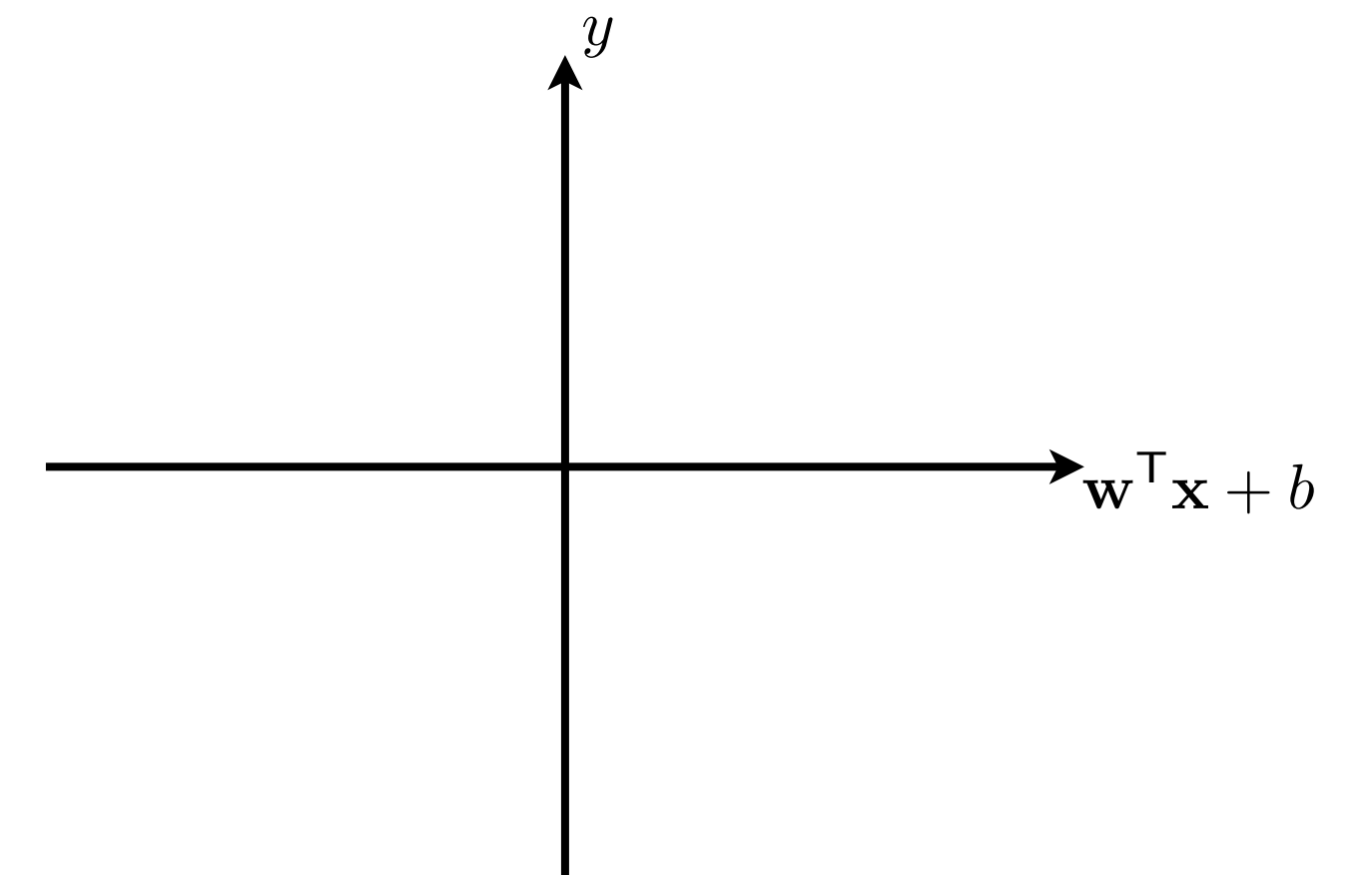
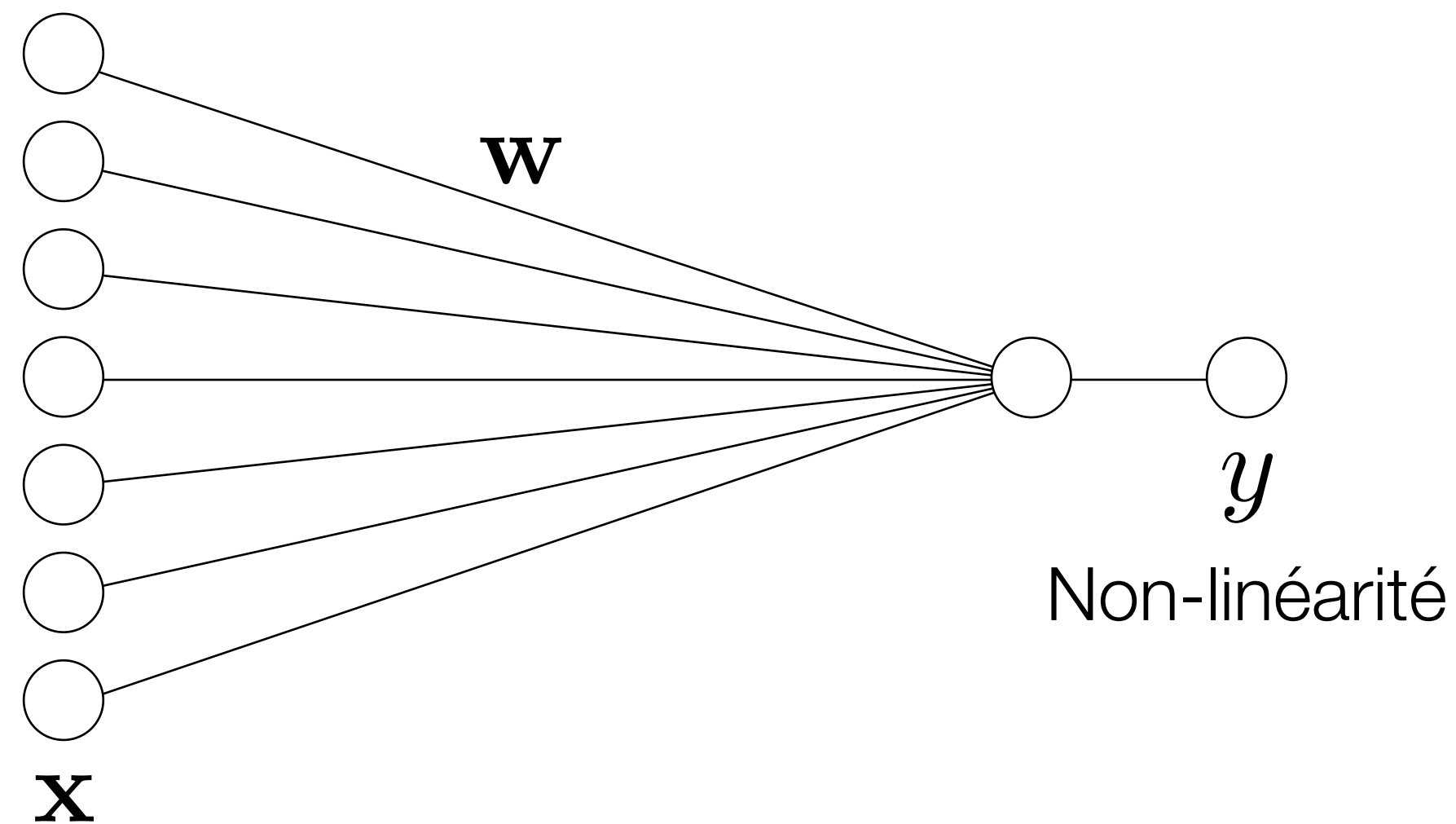
Calculs dans un perceptron

Entrée

Sortie



Calculs dans un perceptron



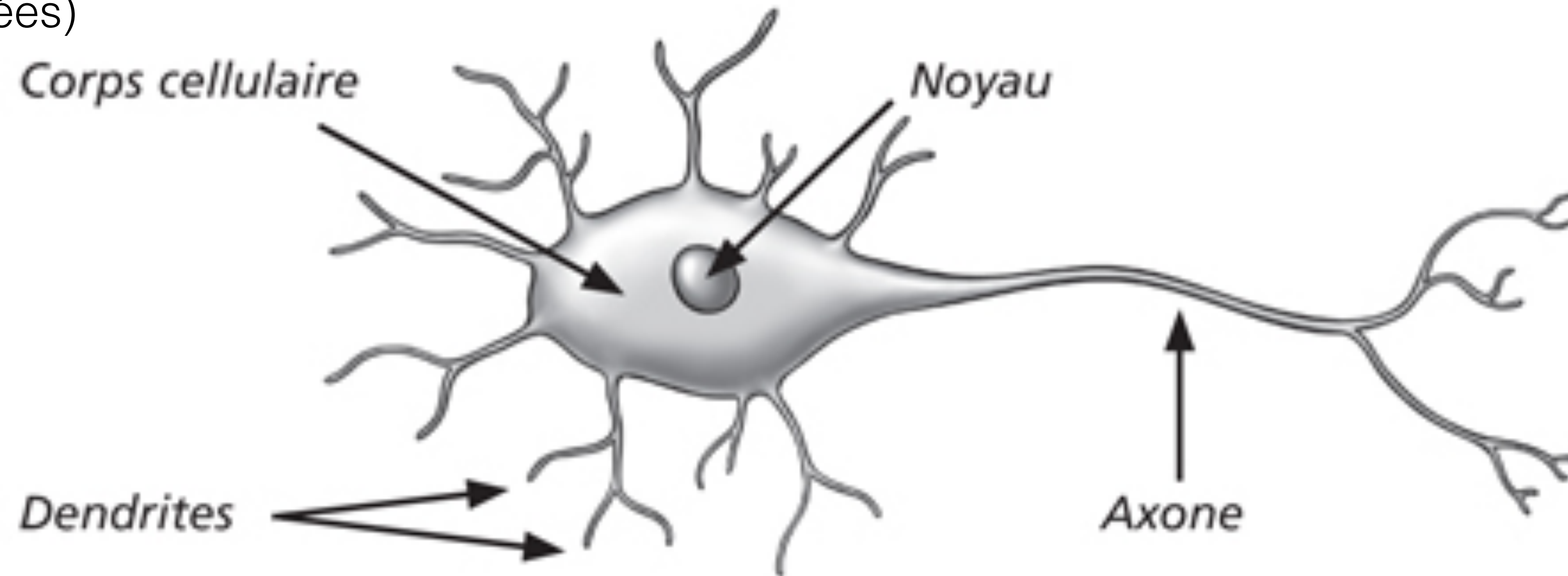
$$y = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$$

Ici, le perceptron prédit une classe (1, -1) pour chaque entrée $\mathbf{x}$.

Inspiration : neurone biologique

$$y = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$$

1. signaux provenant
d'autres neurones
(entrées)



3. signal transmis
si potentiel suffisant
(activation)

2. accumulation (ou soustraction)
de potentiel électrique
(somme pondérée)

Partis pour la gloire

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo
of Computer Designed to
Read and Grow Wiser

WASHINGTON, July 7 (UPI)—The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's \$2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of \$100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human be-

ings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

Entraîner le perceptron

- Initialiser les poids $\mathbf{w}$ aléatoirement
- Itérer sur les exemples d'entraînement plusieurs fois
- Pour chaque exemple d'entraînement $\mathbf{x}$ avec étiquette désirée y :
 - Classifier l'exemple

$$y' = \text{sign}(\mathbf{w}^T \mathbf{x}) + b$$

- Si l'étiquette prédite n'est pas la bonne, modifier les poids

$$\mathbf{w} \leftarrow \mathbf{w} + \gamma(y - y')\mathbf{x}$$

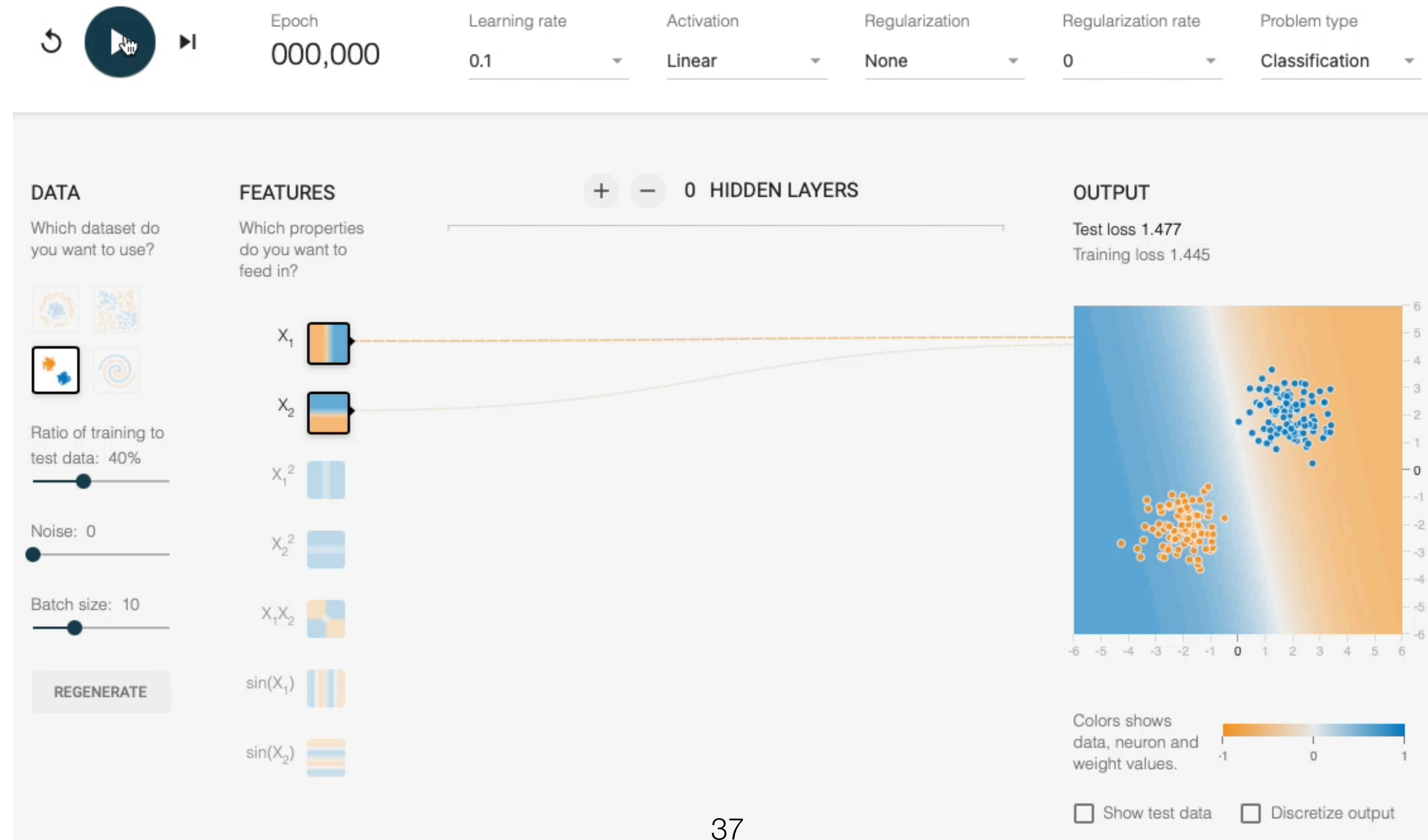
Ici, γ est le taux d'apprentissage.
On le diminue progressivement au fil
de l'entraînement.

Entraîner le perceptron

$$\mathbf{w} \leftarrow \mathbf{w} + \gamma(y - y')\mathbf{x}$$

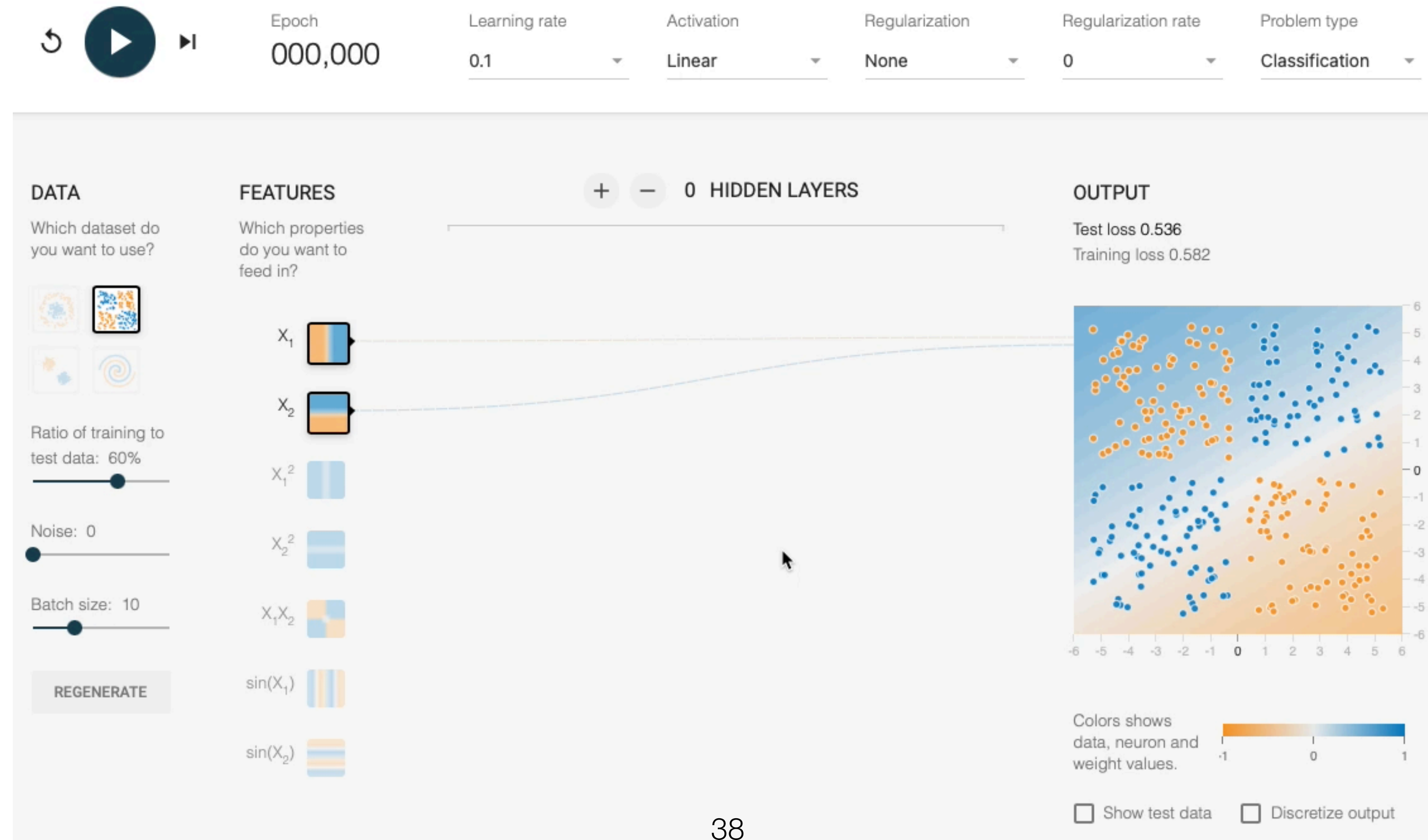
Convergence

- Si les données sont **linéairement séparables**, converge vers une solution parfaite

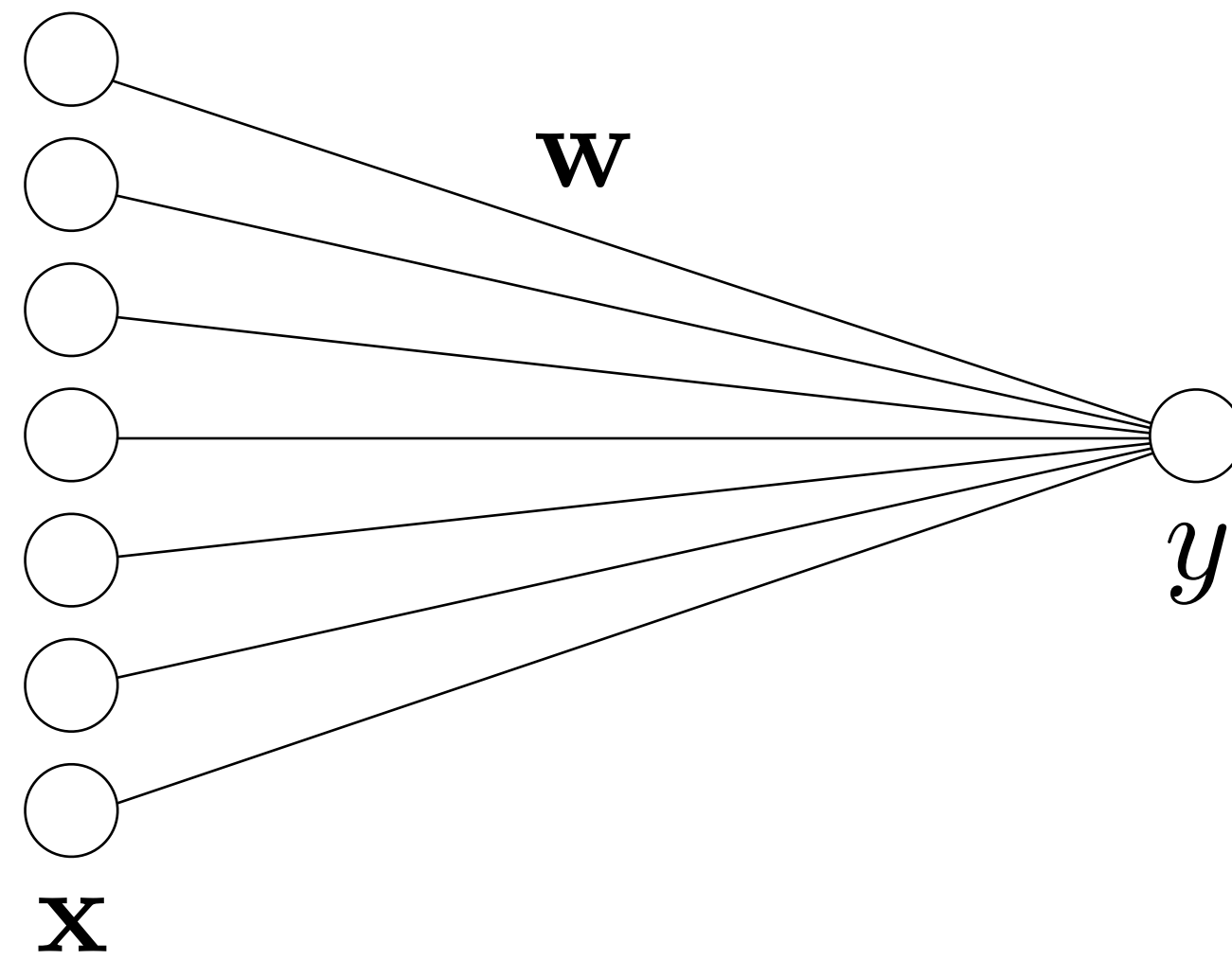


Convergence

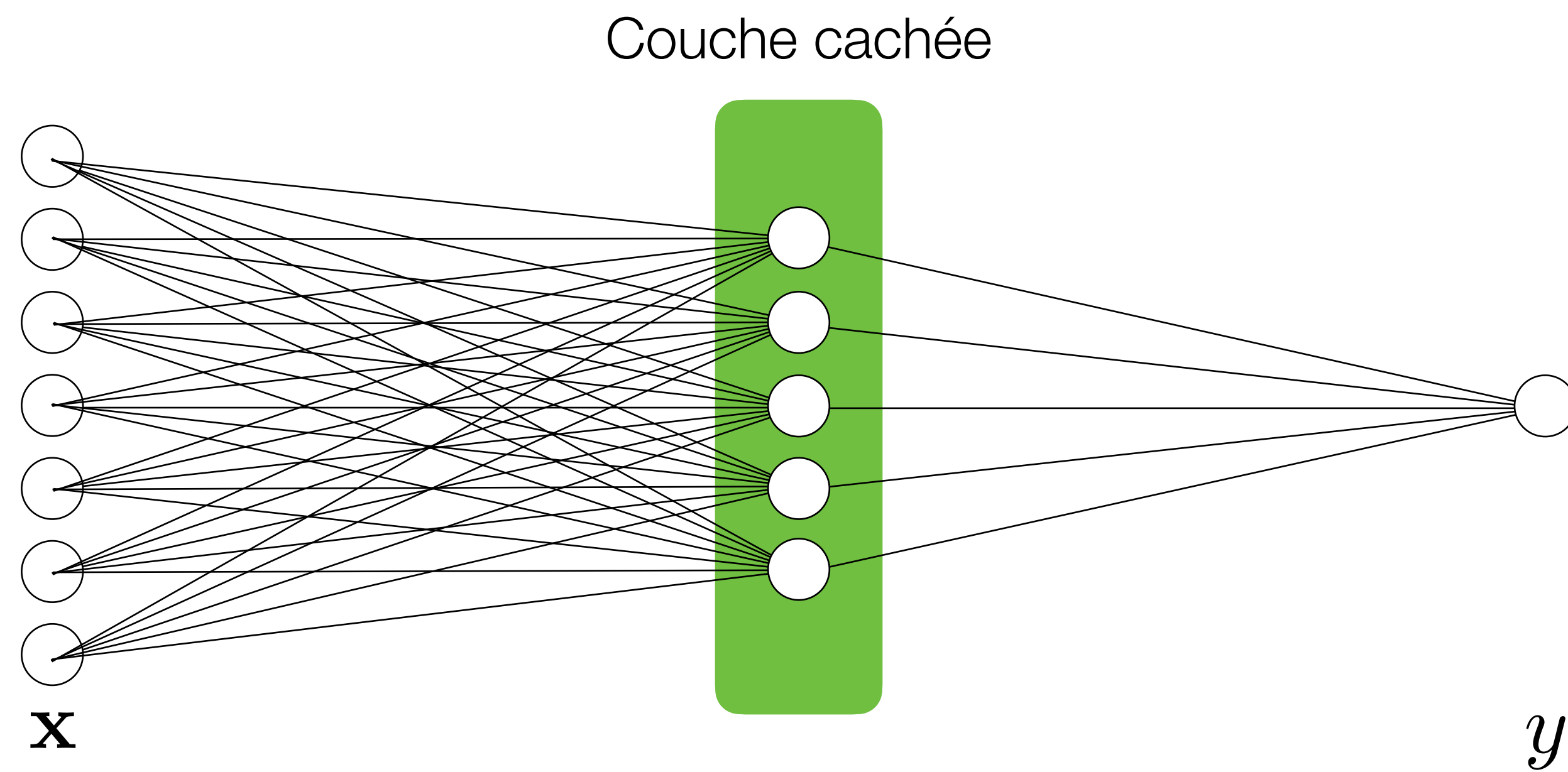
- Si les données ne sont pas **linéairement séparables**, converge vers une solution qui minimise l'erreur



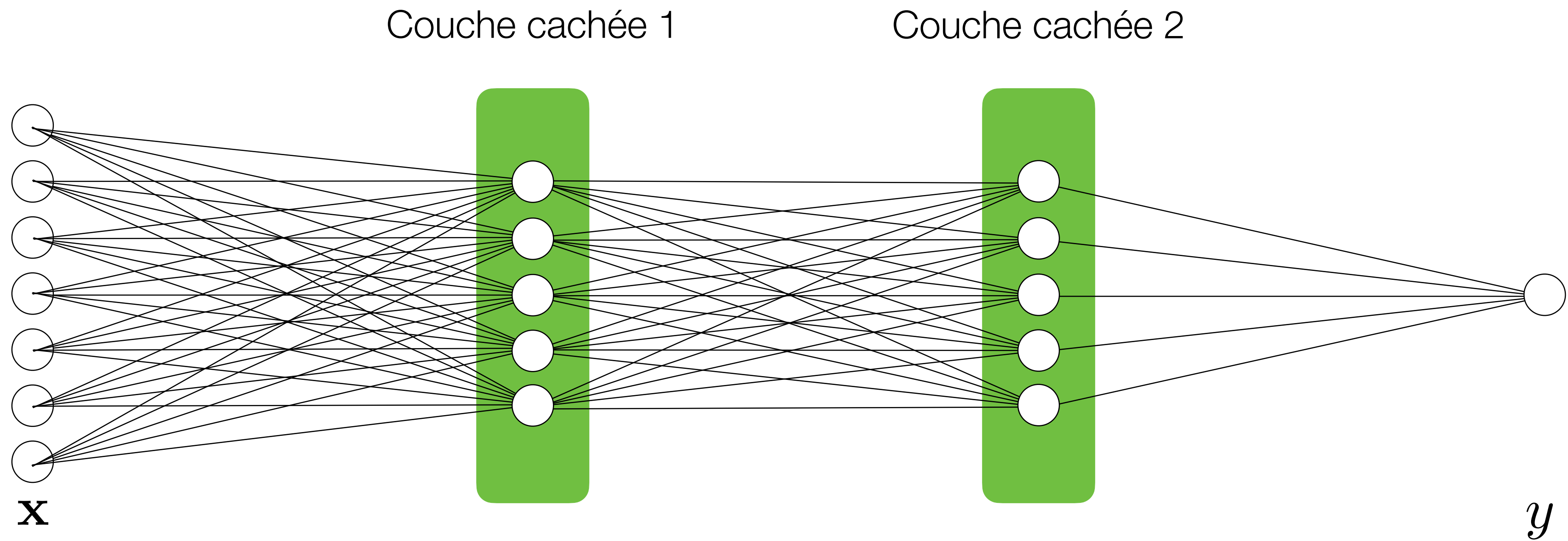
Non-linéaire?



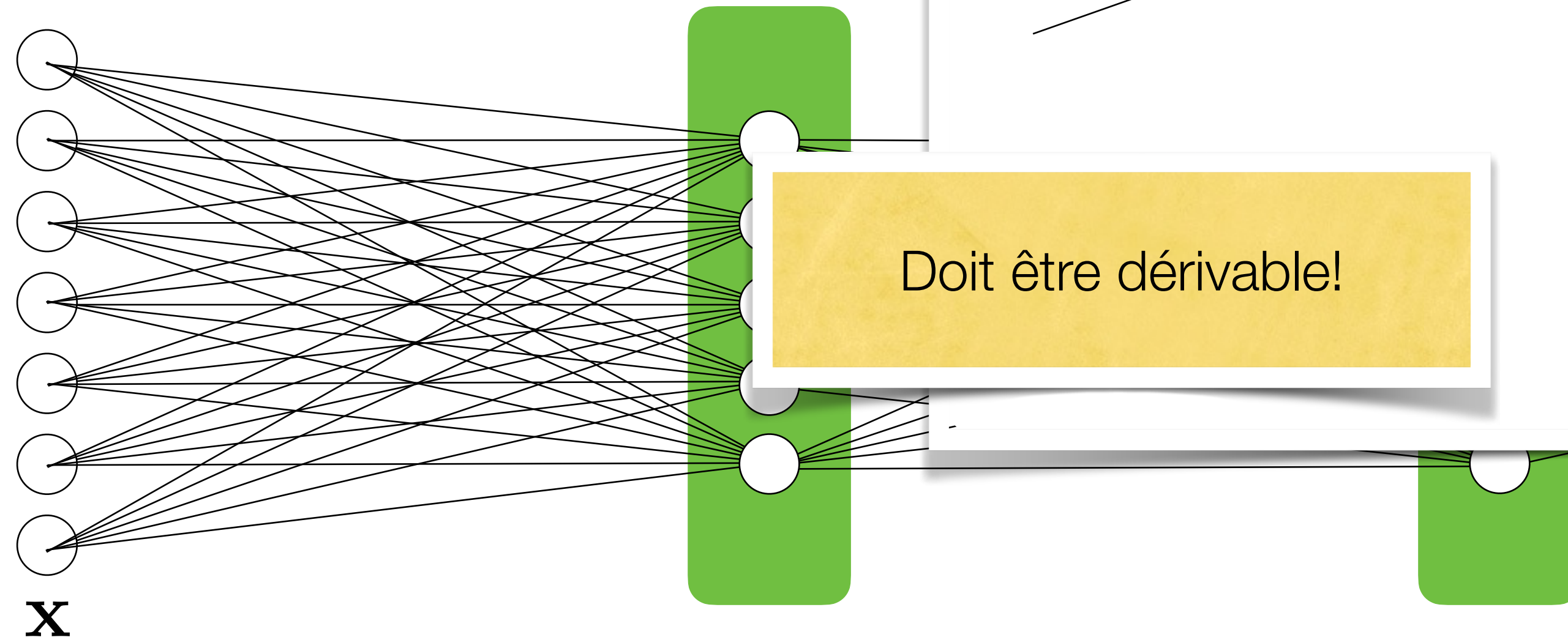
Non-linéaire?



Non-linéaire?

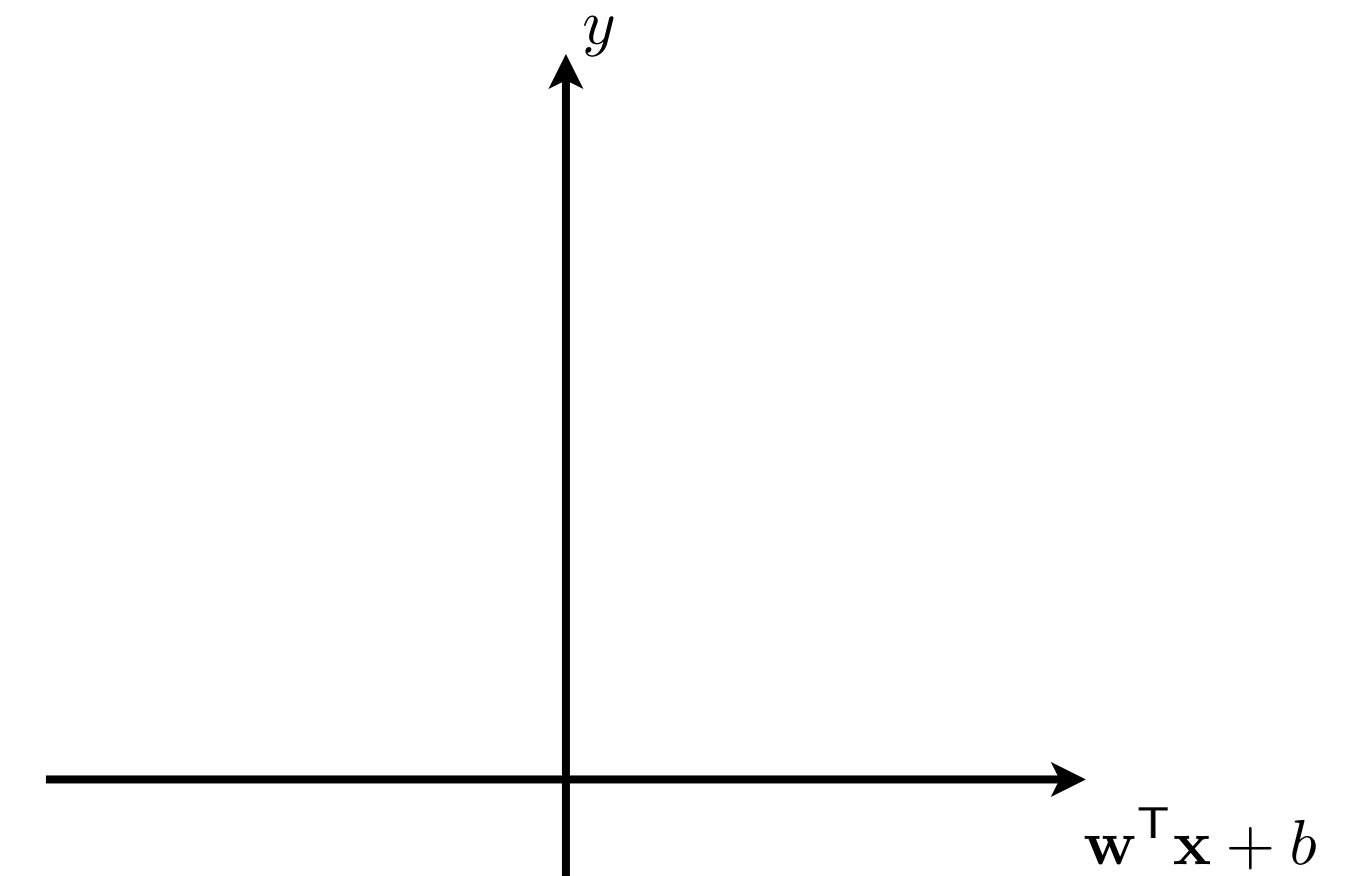


Non-linéaire?



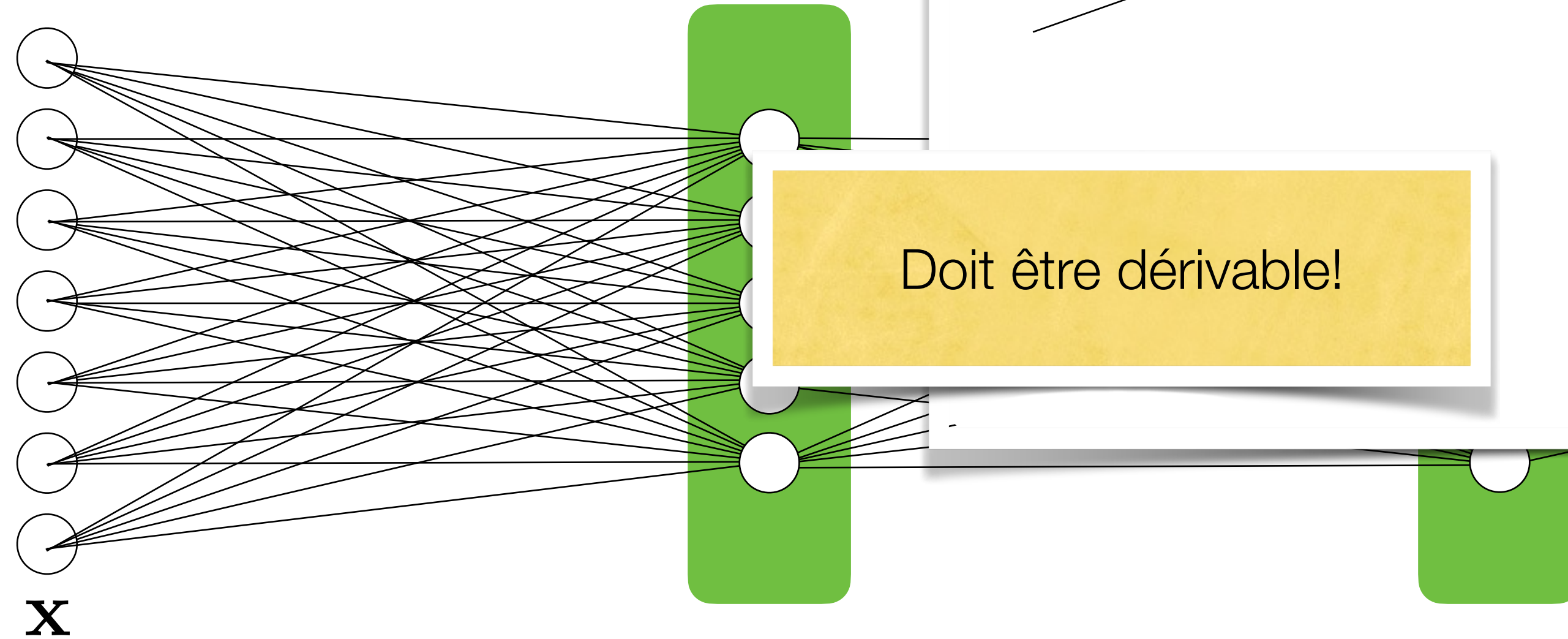
$$y = \text{sigmoid}(\mathbf{w}^T \mathbf{x} + b)$$

$$\text{sigmoid}(t) = \frac{1}{1 + e^{-t}}$$



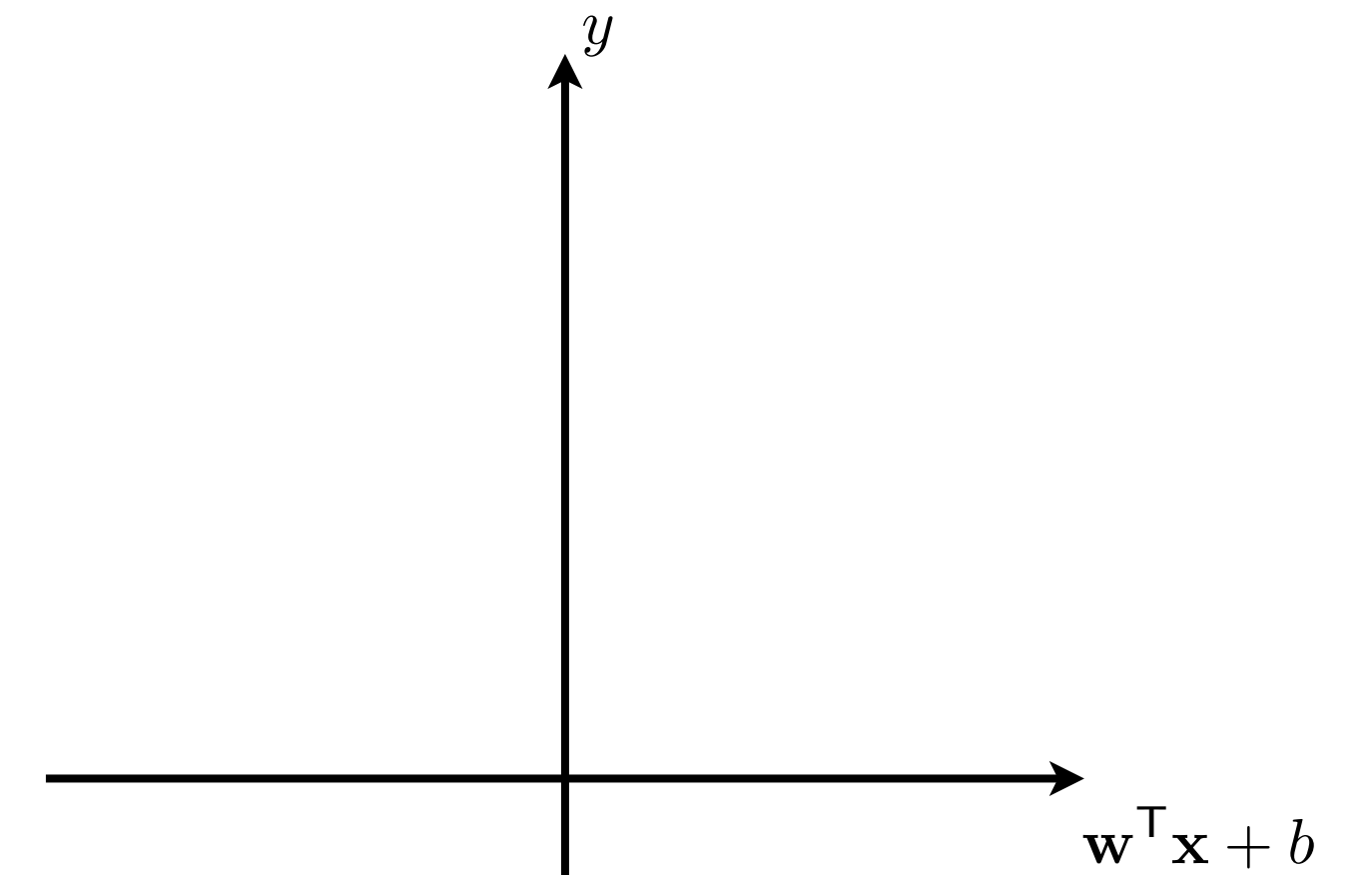
y

Non-linéaire?



$$y = \text{ReLU}(\mathbf{w}^T \mathbf{x} + b)$$

$$\text{ReLU}(t) = \max(0, t)$$



Entraîner le perceptron multi-couches

- Initialiser les poids $\mathbf{w}$ aléatoirement
- Itérer sur les exemples d'entraînement plusieurs fois
- Pour chaque exemple d'entraînement $\mathbf{x}$ avec étiquette désirée y :
 - Classifier l'exemple et calculer une perte

$$\mathcal{L}(\mathbf{x}, y; \mathbf{w})$$

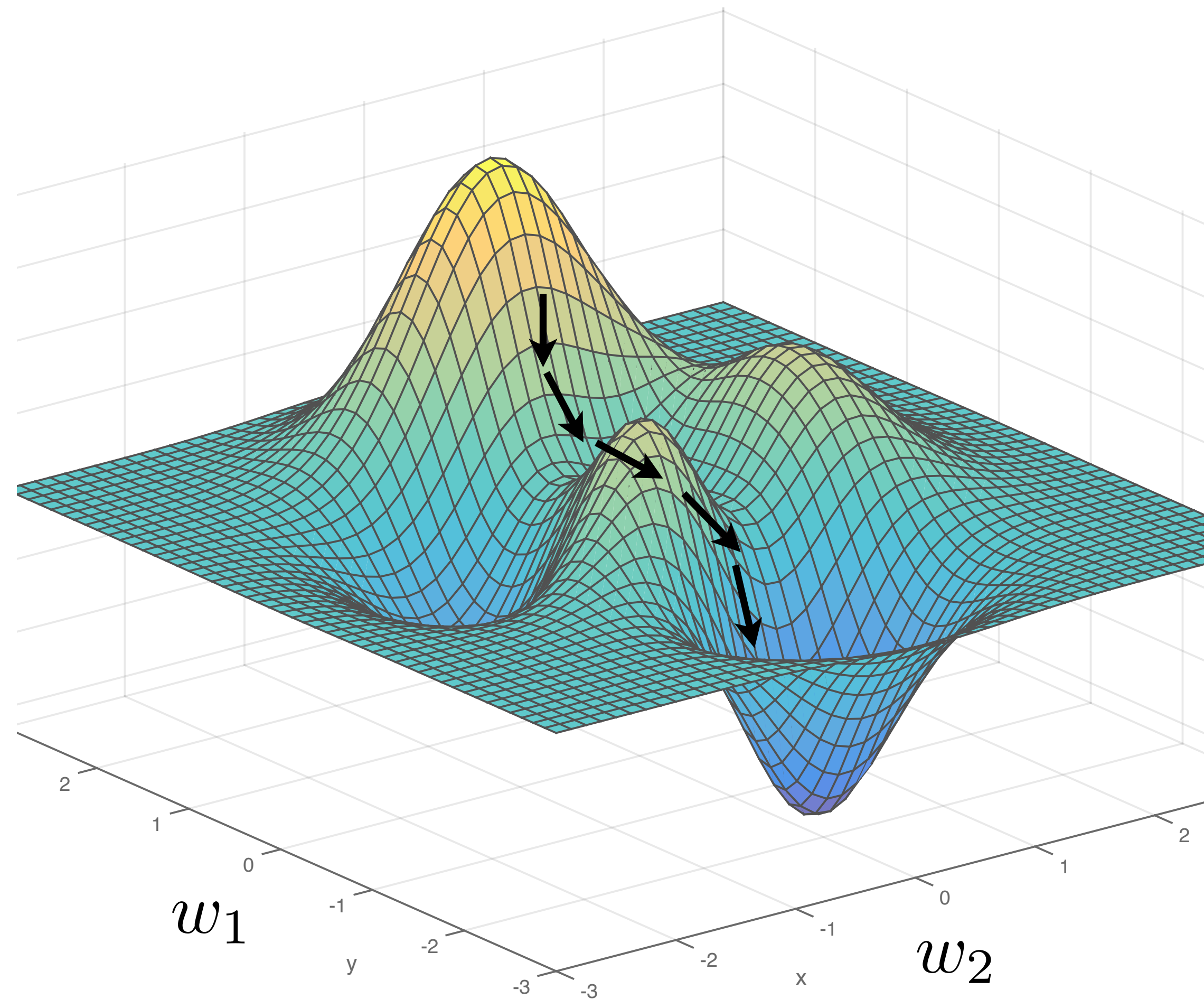
- Si l'étiquette prédite n'est pas la bonne, modifier les poids

$$\mathbf{w} \leftarrow \mathbf{w} - \gamma \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$$

Entraîner le perceptron multi-couches

$$\mathbf{w} \leftarrow \mathbf{w} - \gamma \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$$

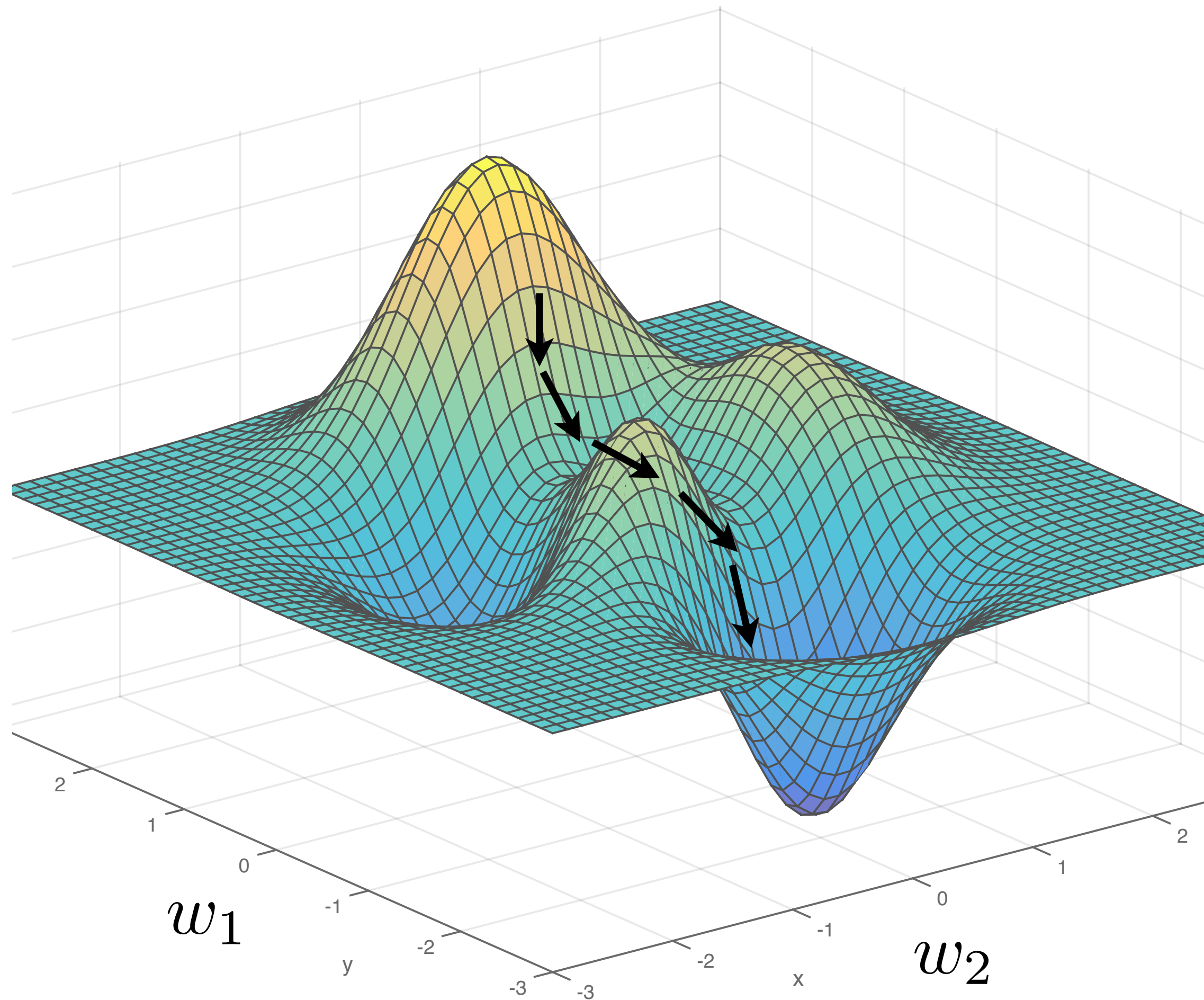
$\mathcal{L}(\mathbf{x}, y; \mathbf{w})$



Entraîner le perceptron multi-couches

$$\mathbf{w} \leftarrow \mathbf{w} - \gamma \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$$

$\mathcal{L}(\mathbf{x}, y; \mathbf{w})$



Descente du gradient

Le gradient de la fonction de perte par rapport à chaque poids du réseau est calculé avec la dérivation en chaîne

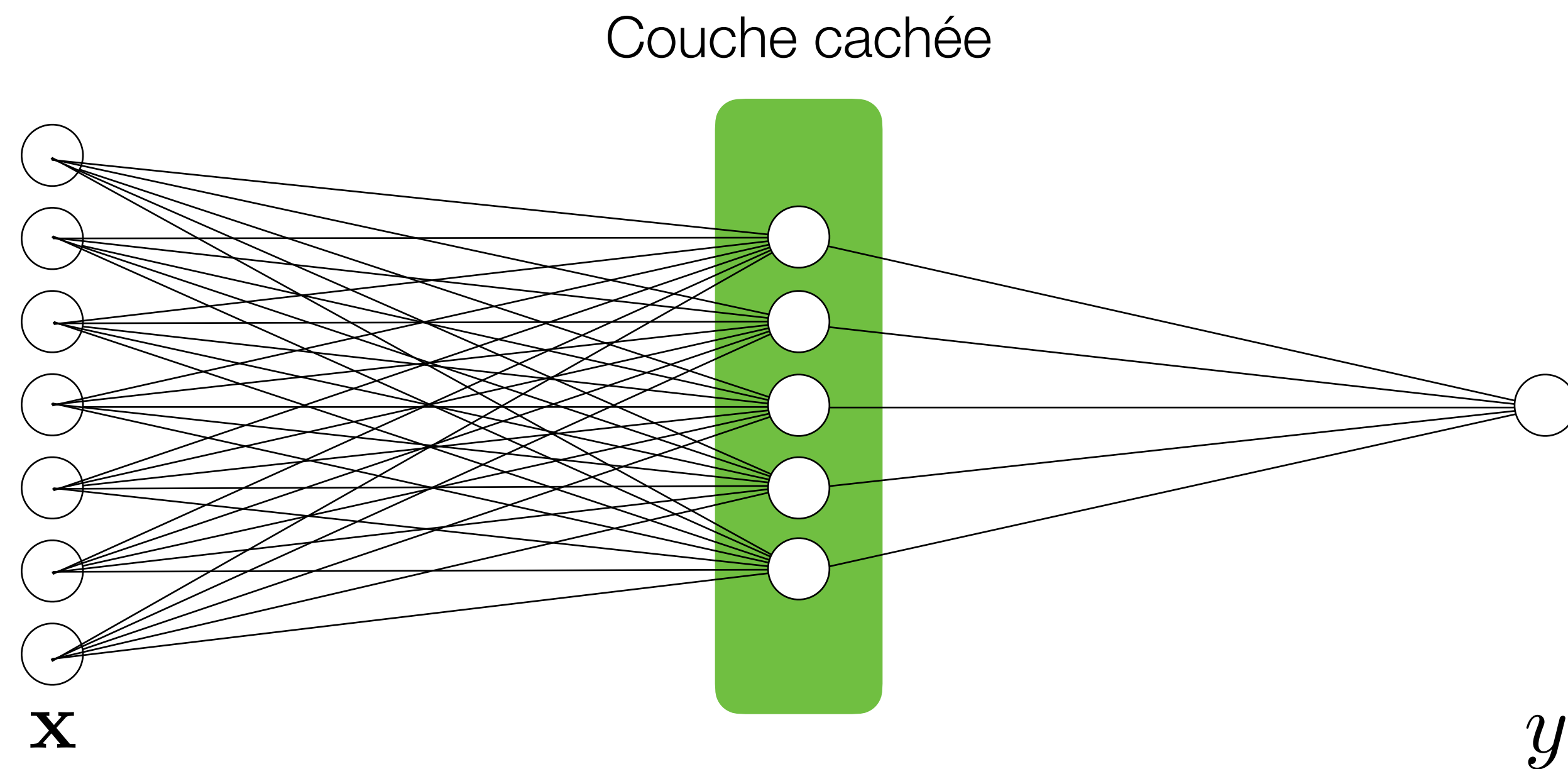
Descente du gradient stochastique

On calcule la mise à jour des poids à partir d'un sous-ensemble aléatoire des données d'entraînement, et on répète!

$$E(\mathbf{w}) = \sum_i \mathcal{L}(\mathbf{x}_i, y_i; \mathbf{w})$$

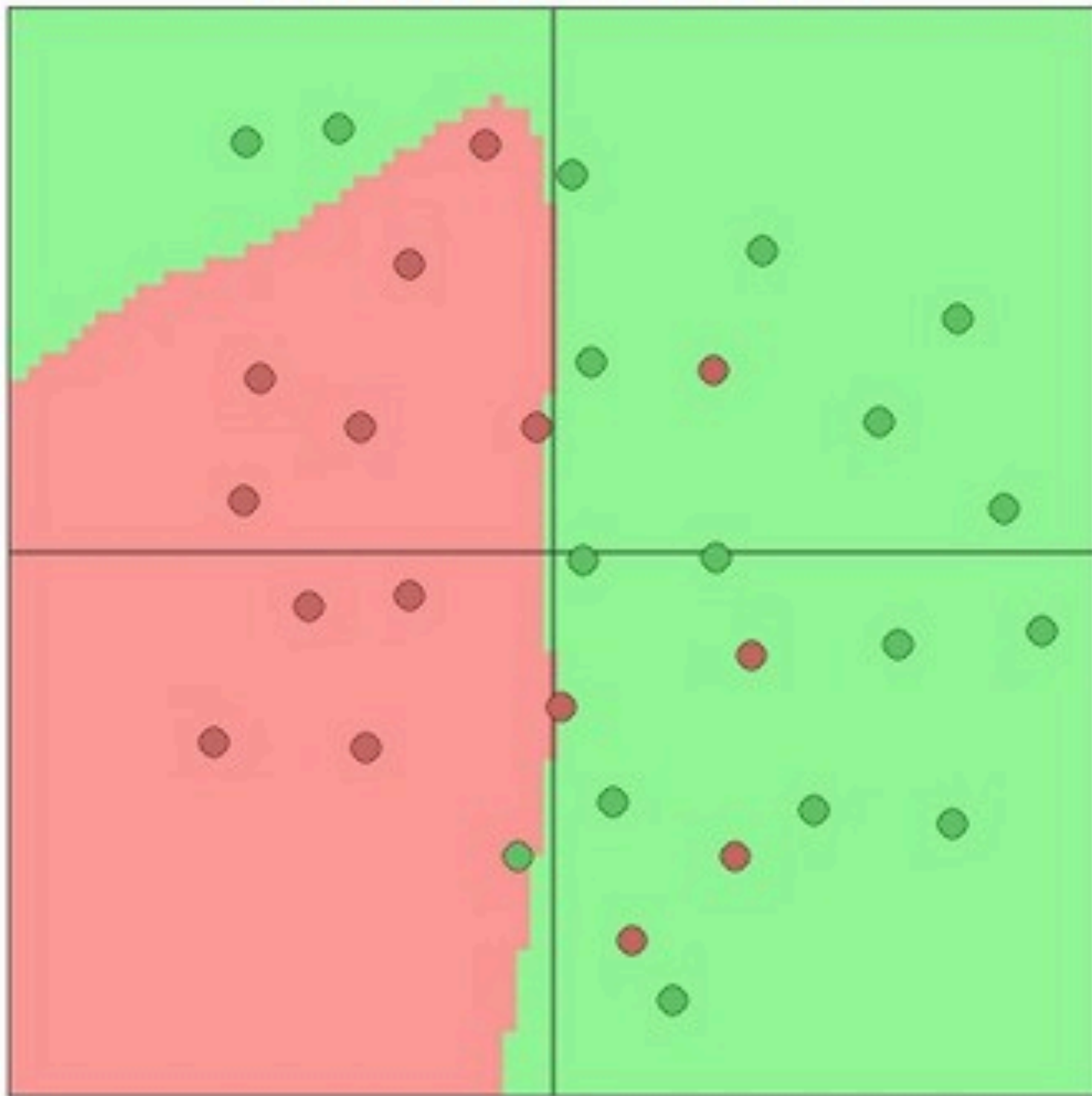
Perceptron à une couche cachée

Les réseaux de neurones à une couche cachée sont des **approximateurs de fonctions** universels.



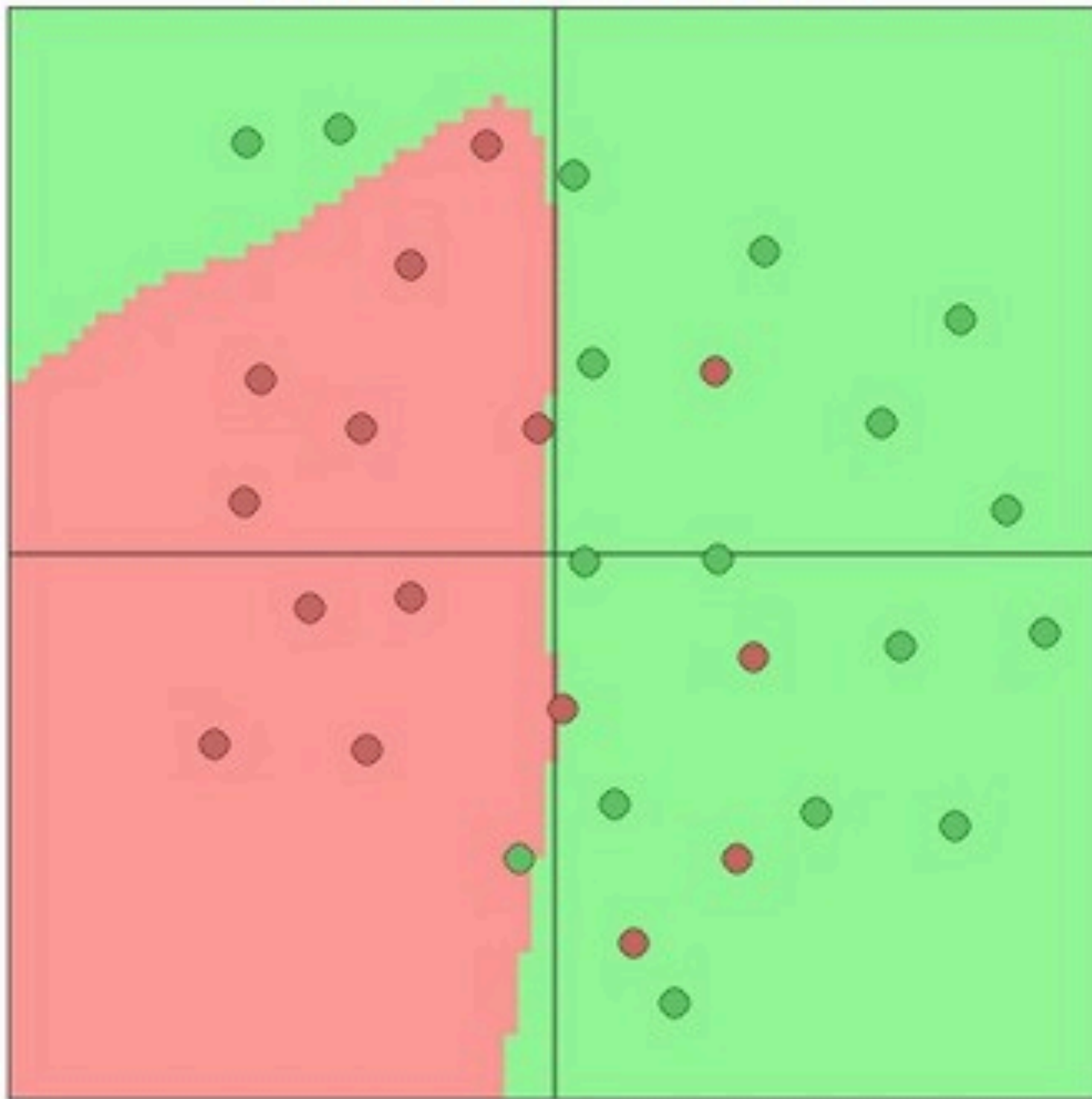
Perceptron à une couche cachée

3 neurones sur la couche cachée

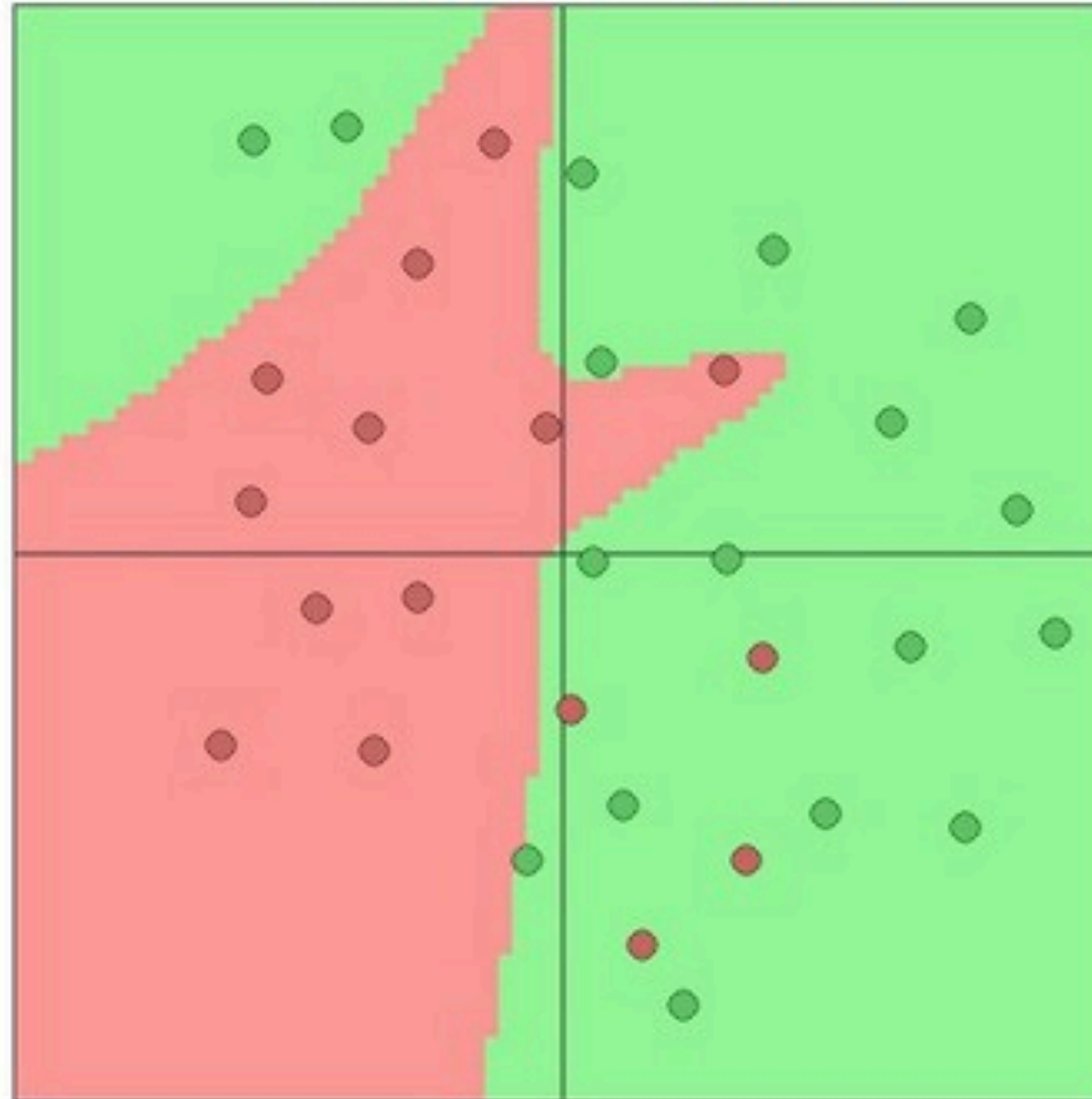


Perceptron à une couche cachée

3 neurones sur la couche cachée

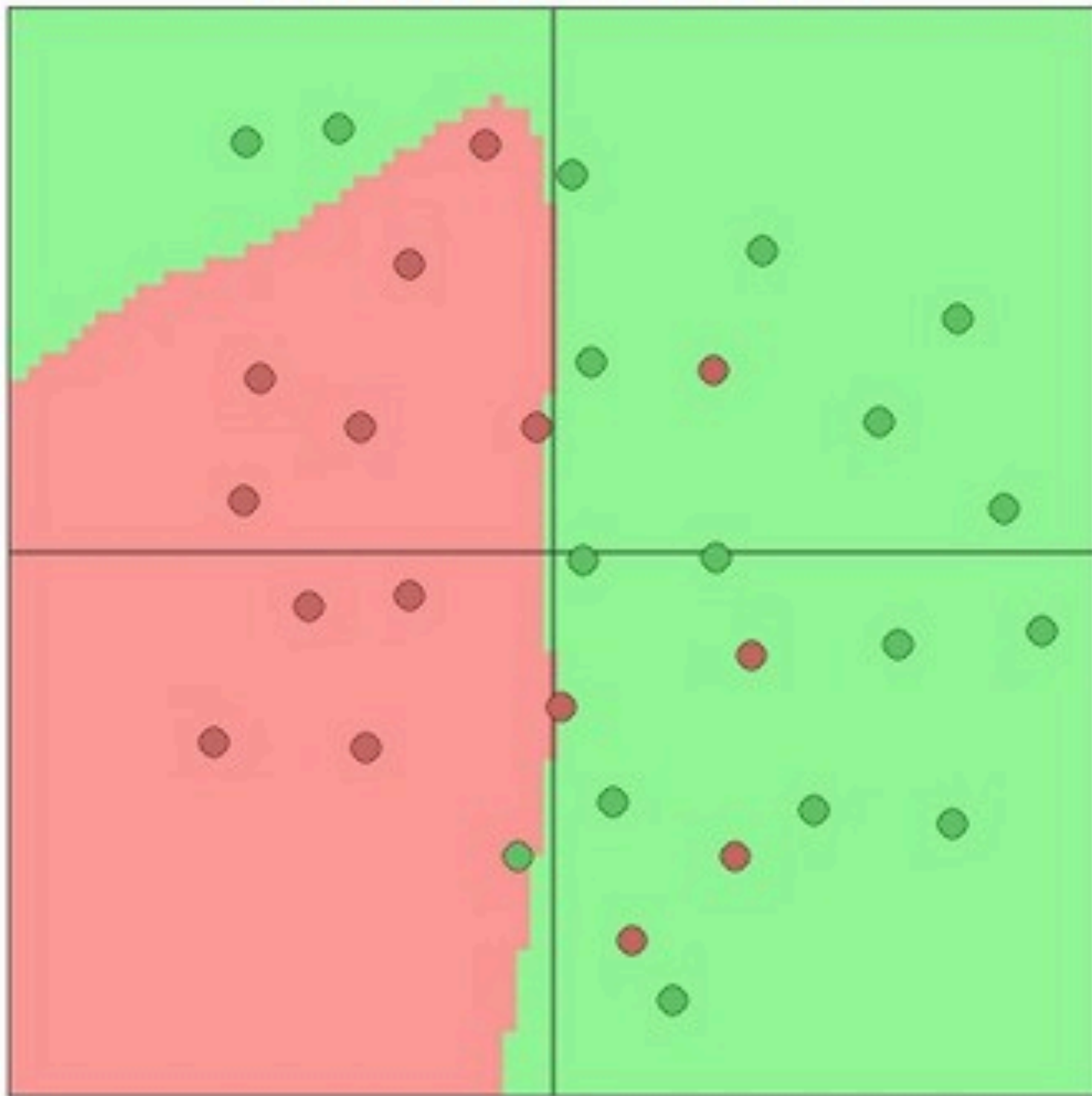


6 neurones sur la couche cachée

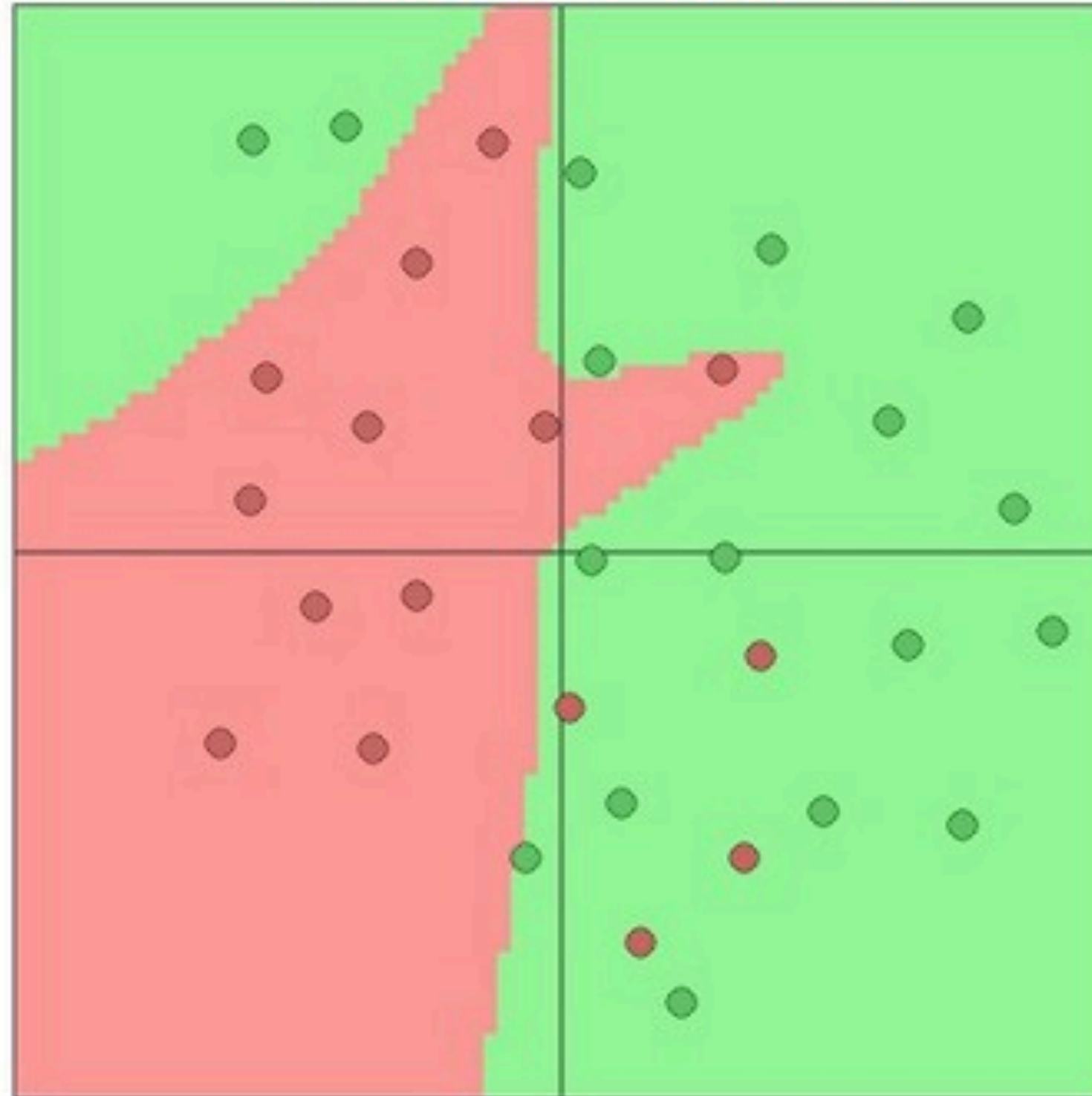


Perceptron à une couche cachée

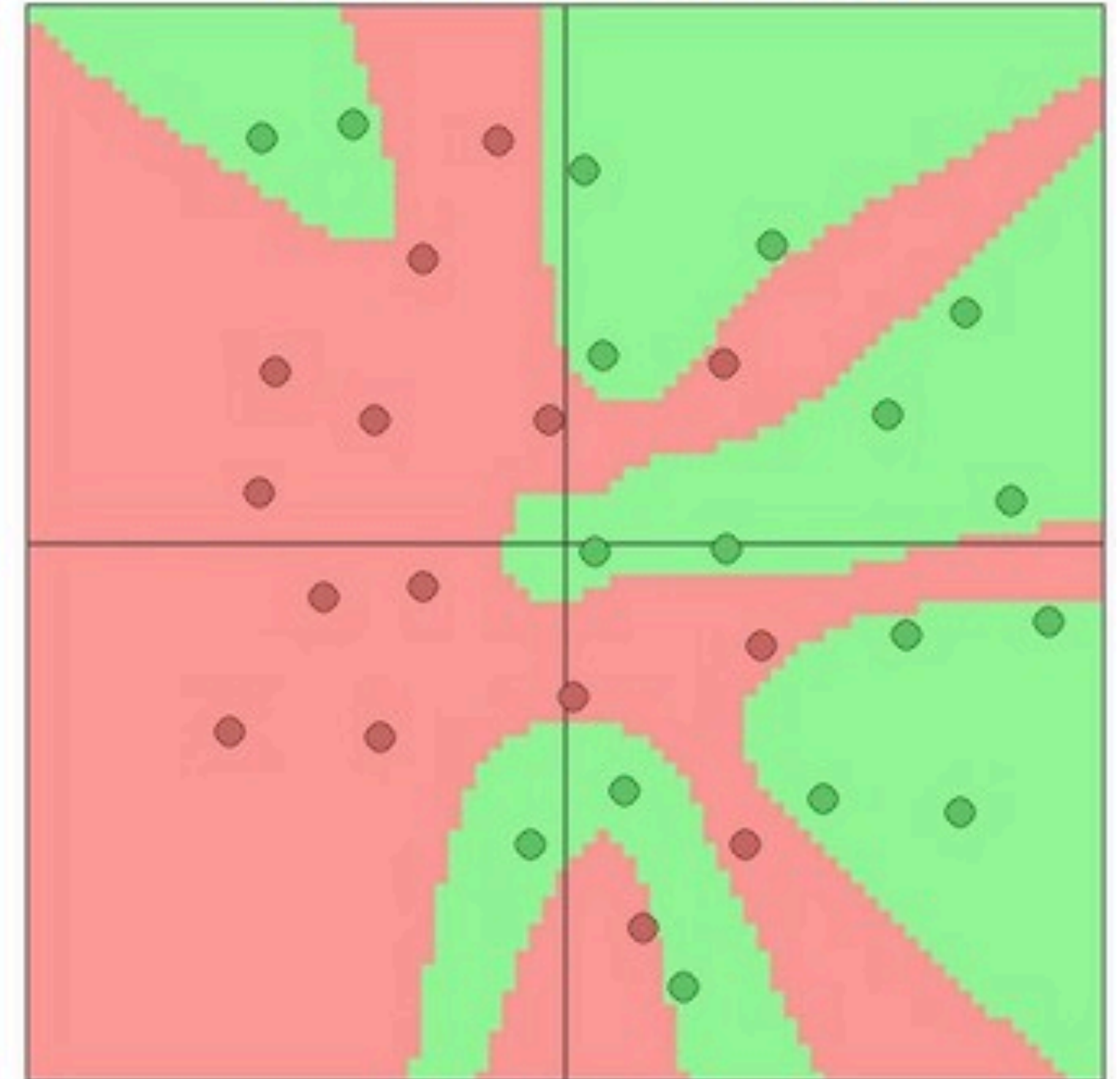
3 neurones sur la couche cachée



6 neurones sur la couche cachée



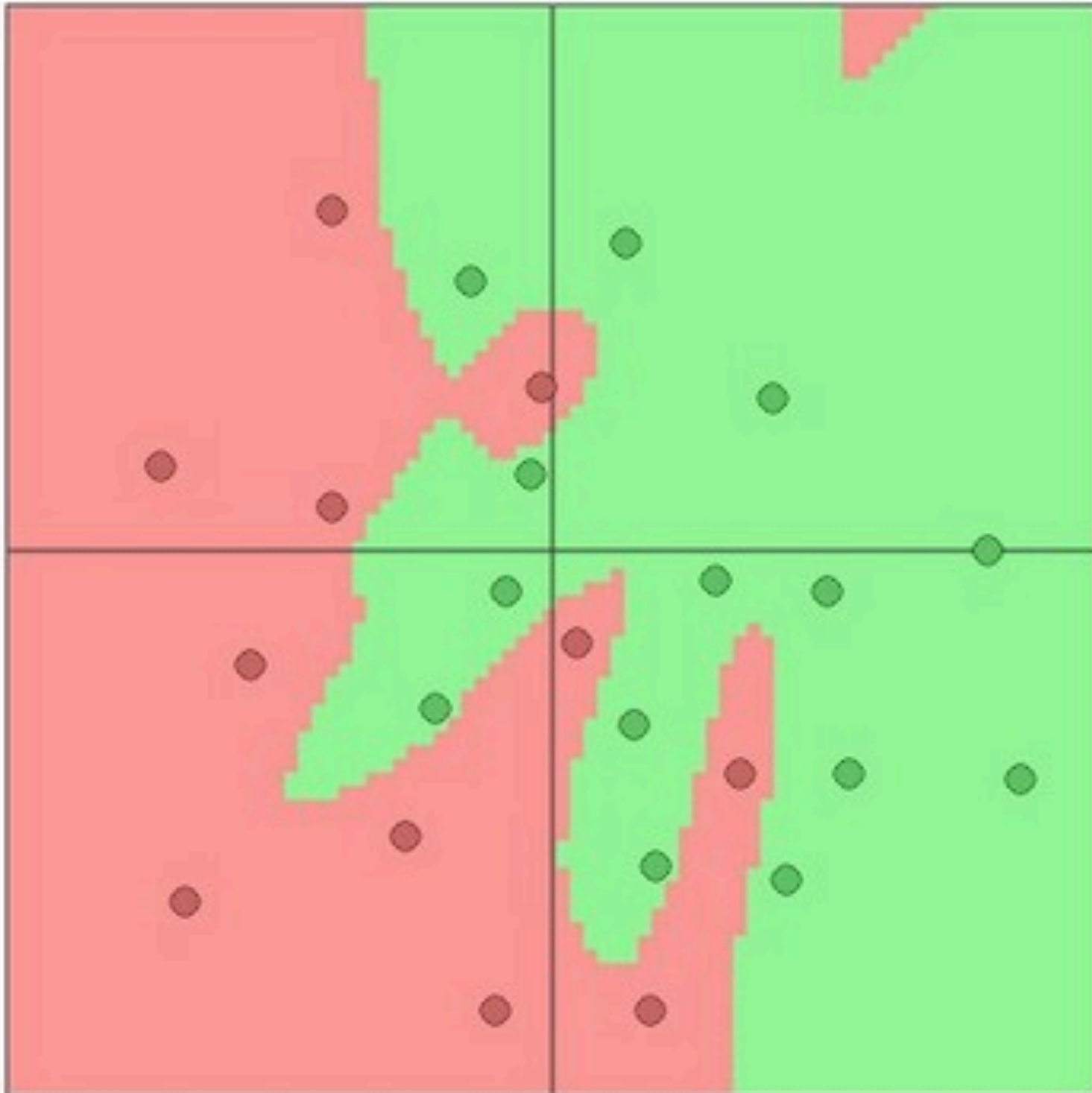
20 neurones sur la couche cachée



Régularisation

$$E(\mathbf{w}) = \sum_i \mathcal{L}(\mathbf{x}_i, y_i; \mathbf{w}) + \lambda ||\mathbf{w}||^2$$

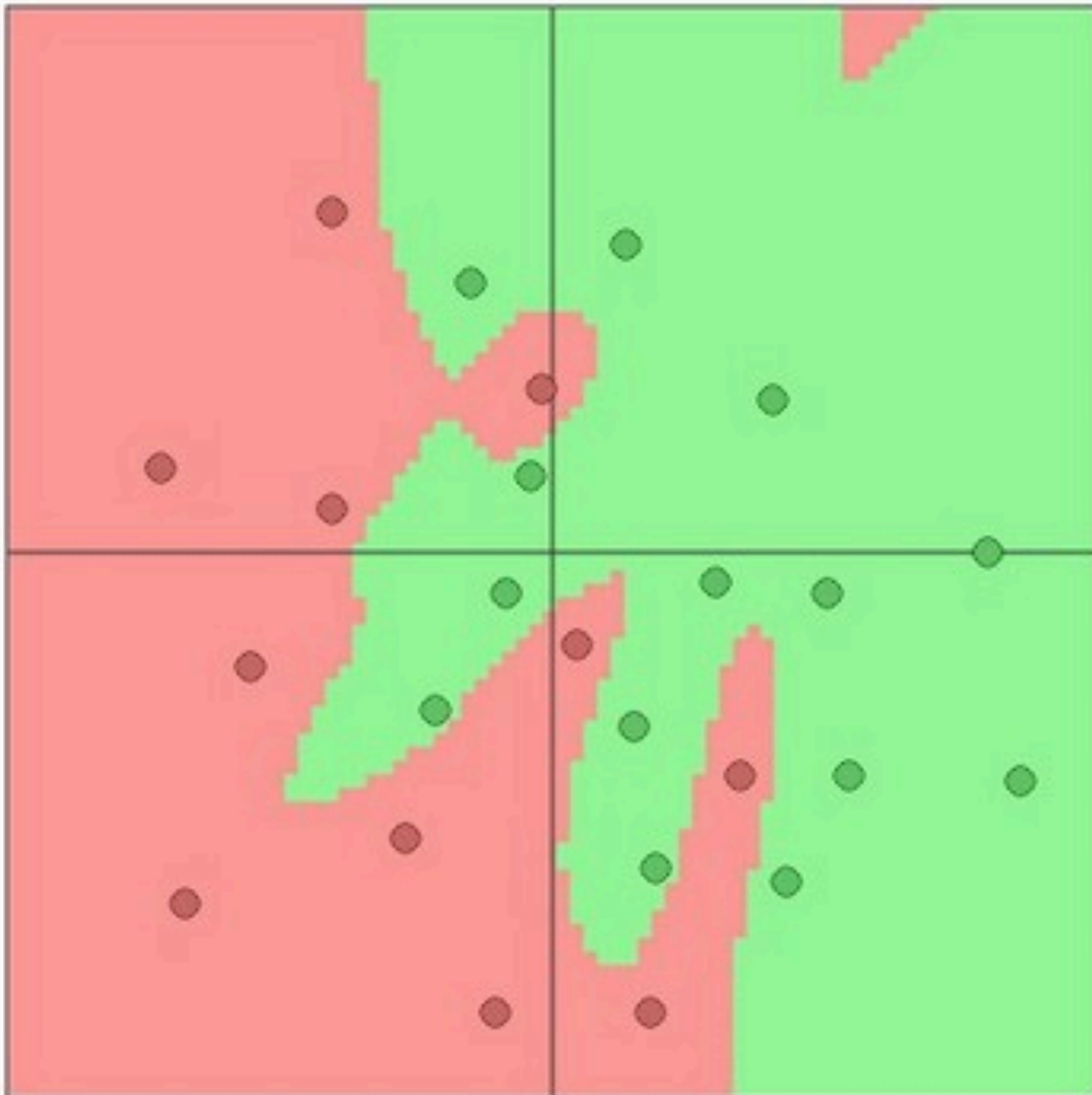
$\lambda = 0.001$



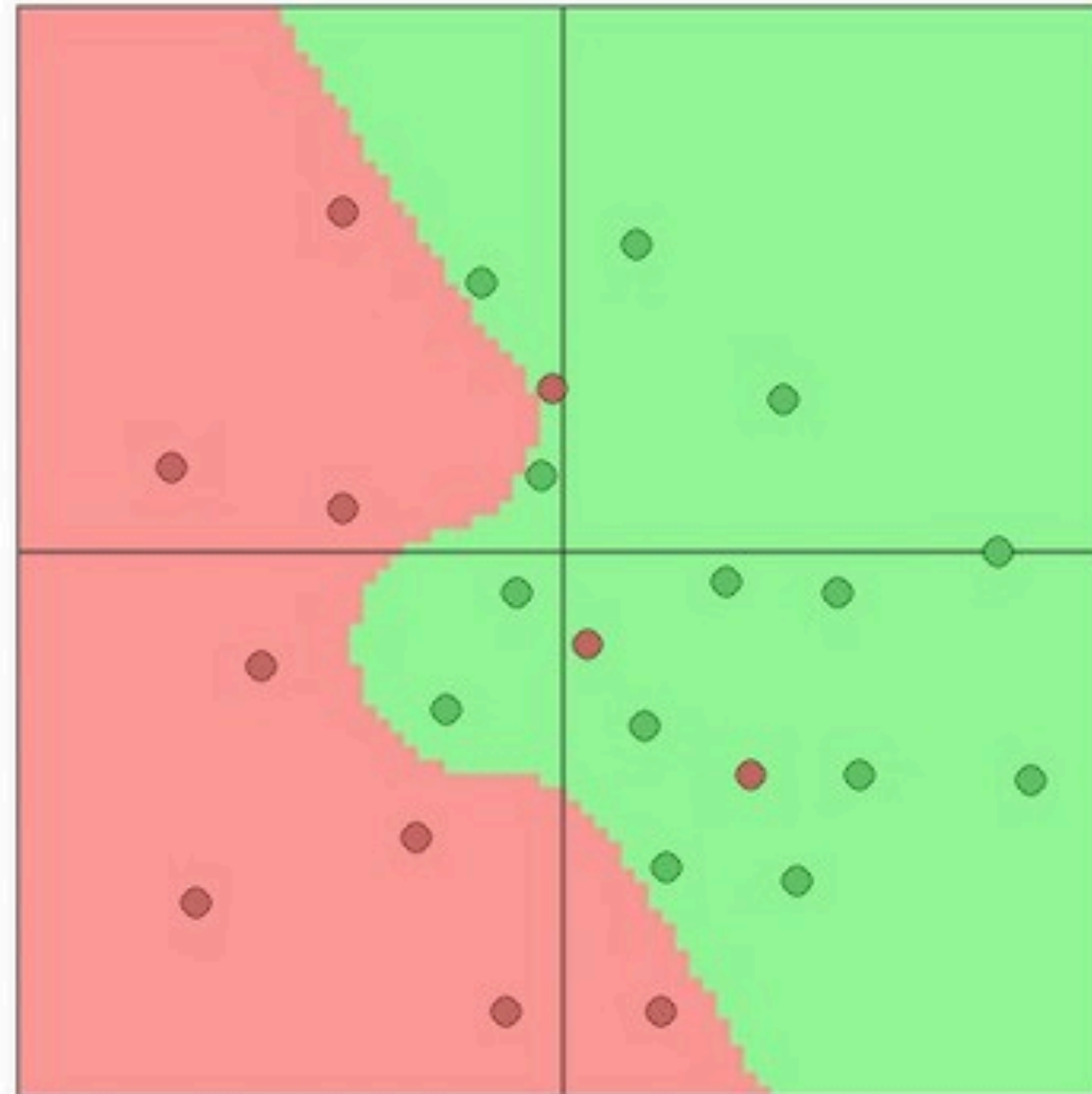
Régularisation

$$E(\mathbf{w}) = \sum_i \mathcal{L}(\mathbf{x}_i, y_i; \mathbf{w}) + \lambda ||\mathbf{w}||^2$$

$\lambda = 0.001$



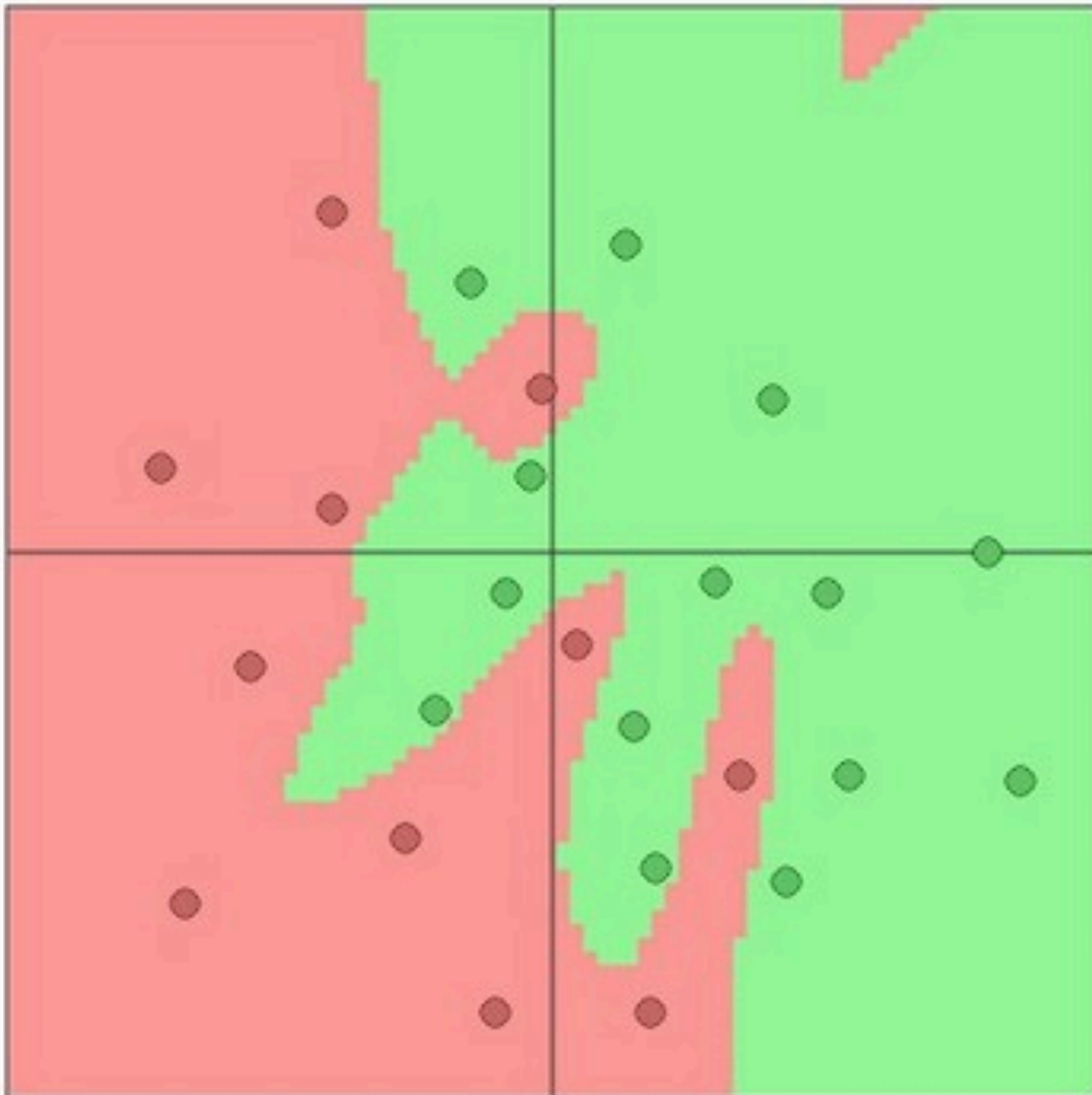
$\lambda = 0.01$



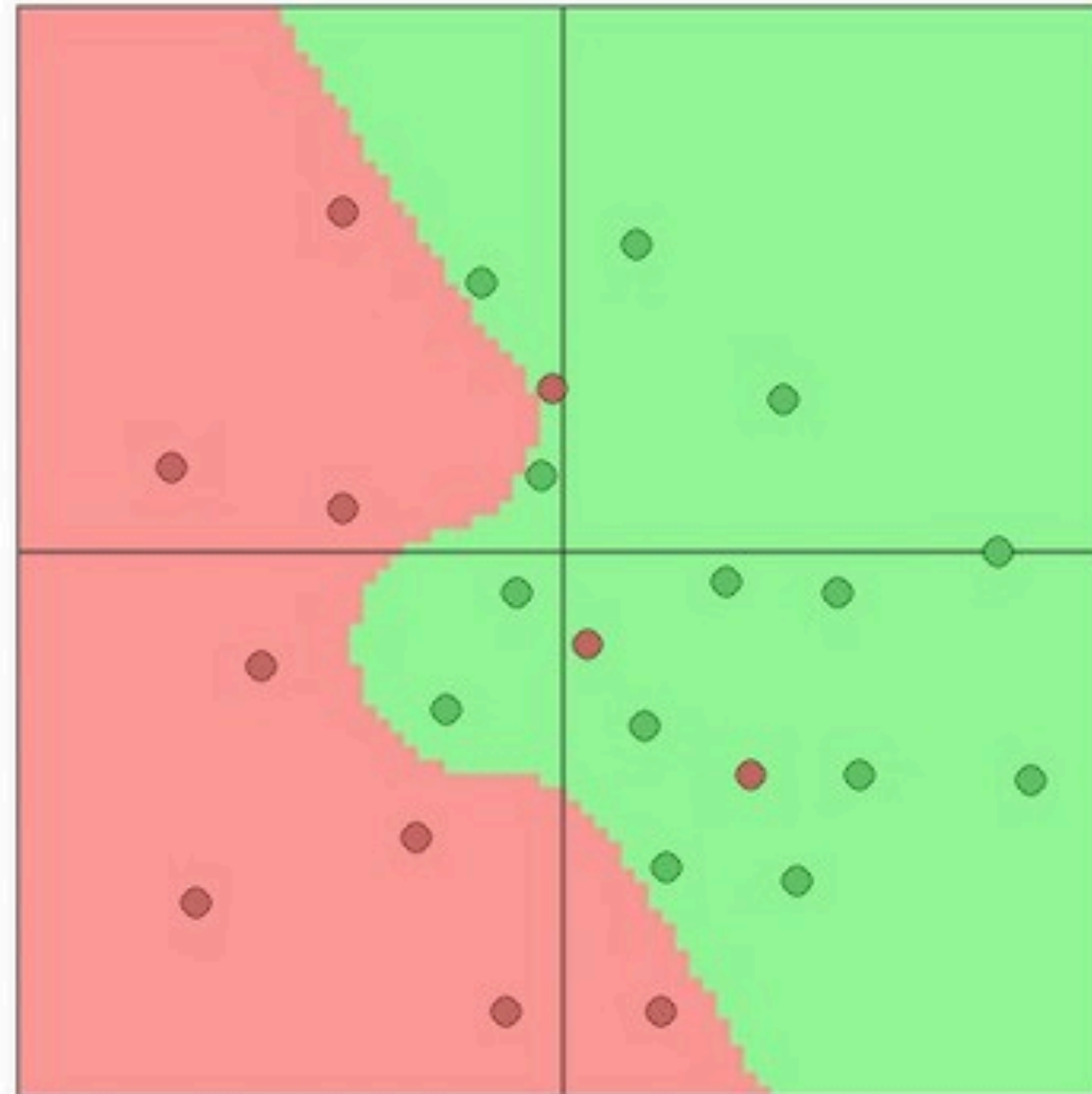
Régularisation

$$E(\mathbf{w}) = \sum_i \mathcal{L}(\mathbf{x}_i, y_i; \mathbf{w}) + \lambda ||\mathbf{w}||^2$$

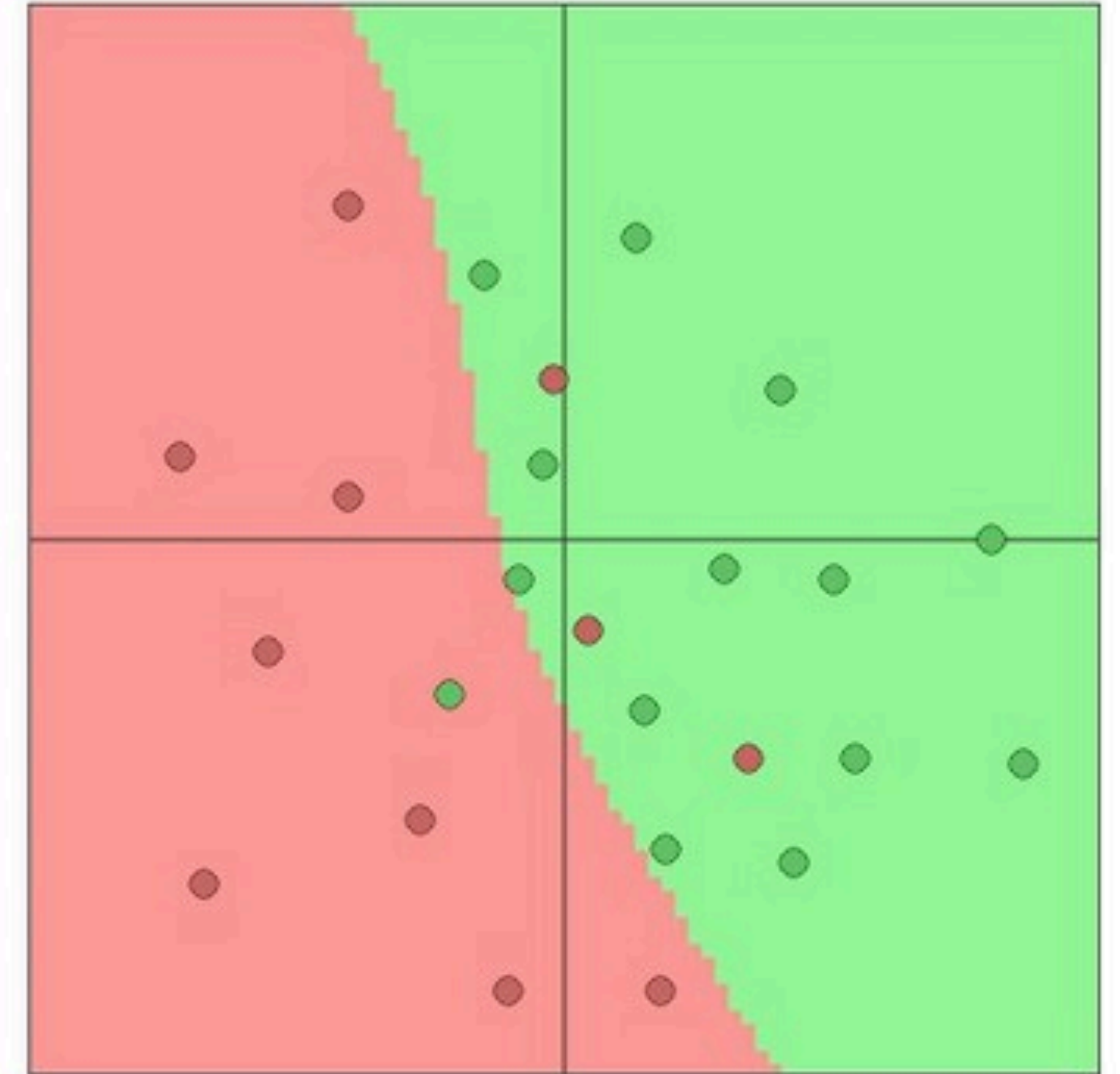
$\lambda = 0.001$



$\lambda = 0.01$

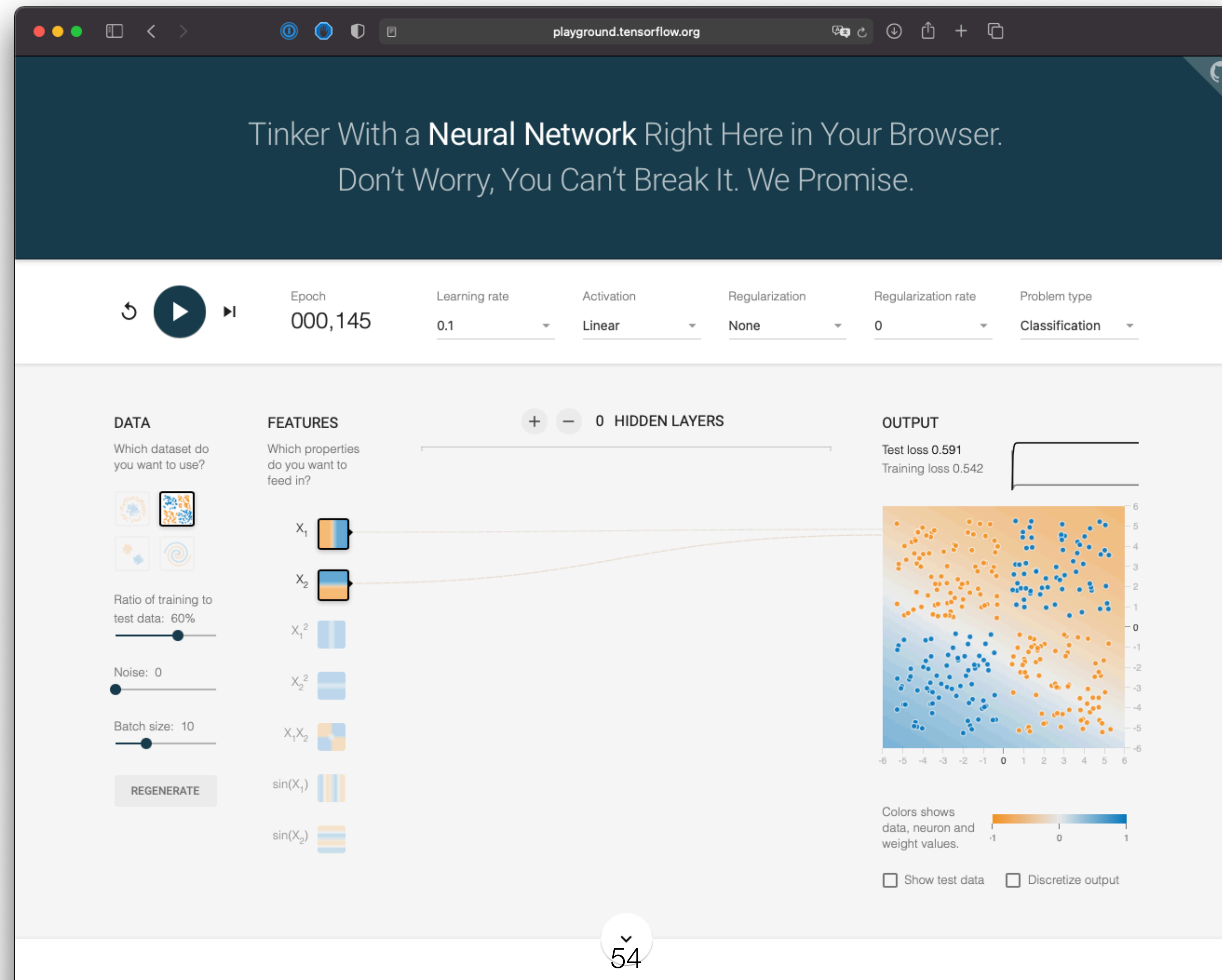


$\lambda = 0.1$



Expérimentez avec le perceptron multi-couches!

<http://playground.tensorflow.org>



Génération d'images par apprentissage profond

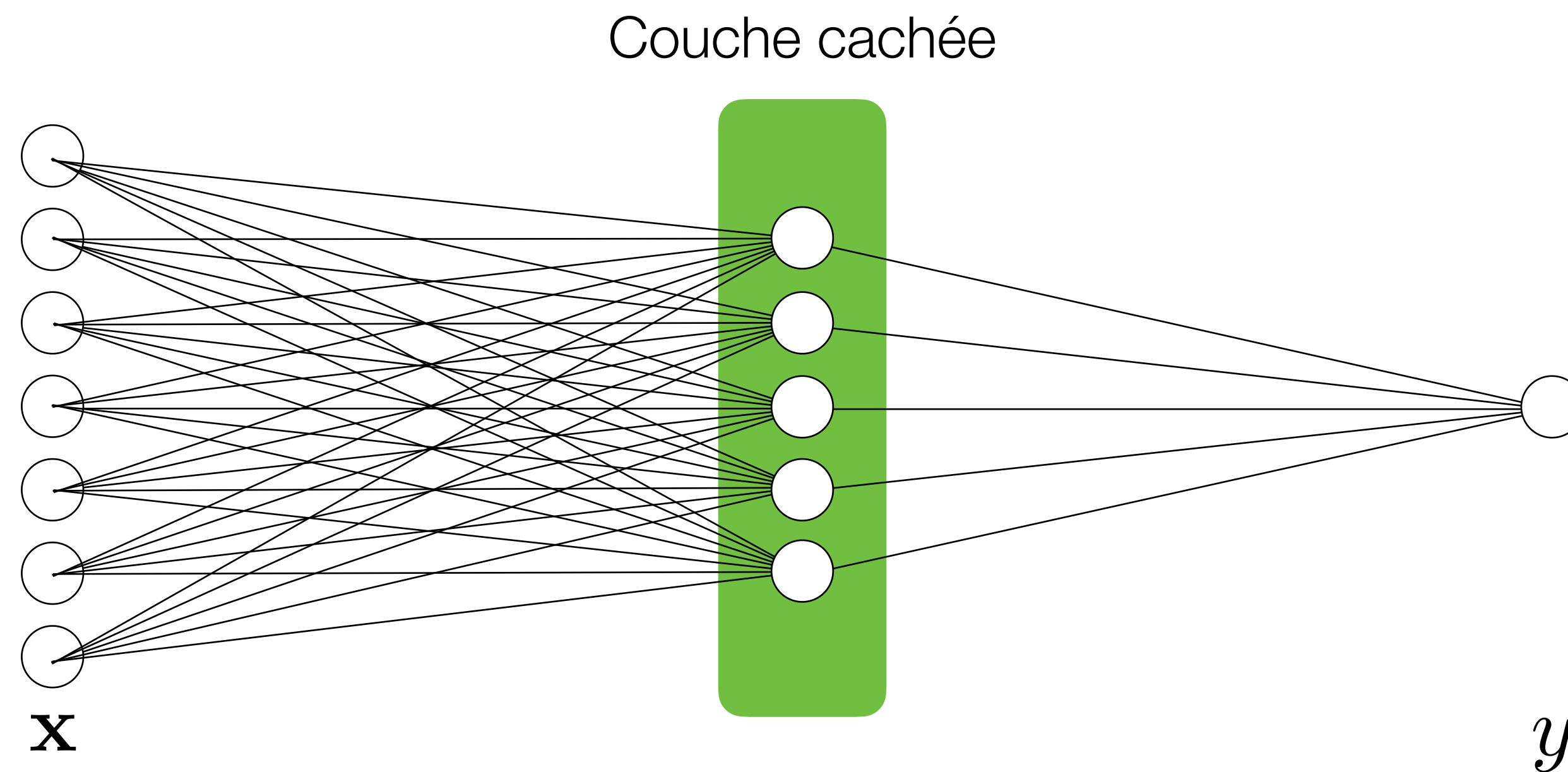
Les réseaux de neurones à convolution (CNN)



GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Perceptron à une couche cachée

Les réseaux de neurones à une couche cachée sont des **approximateurs de fonctions** universels.

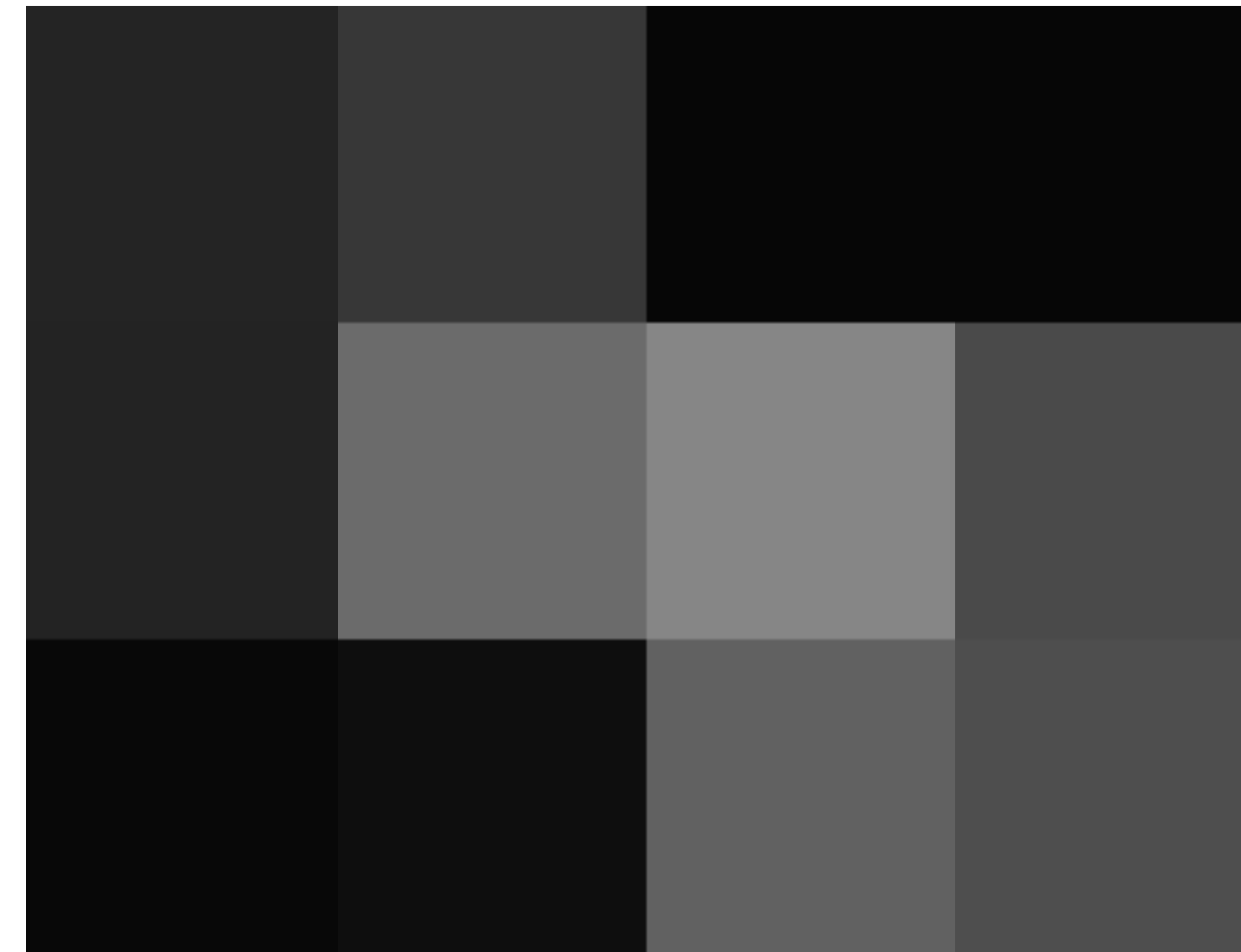
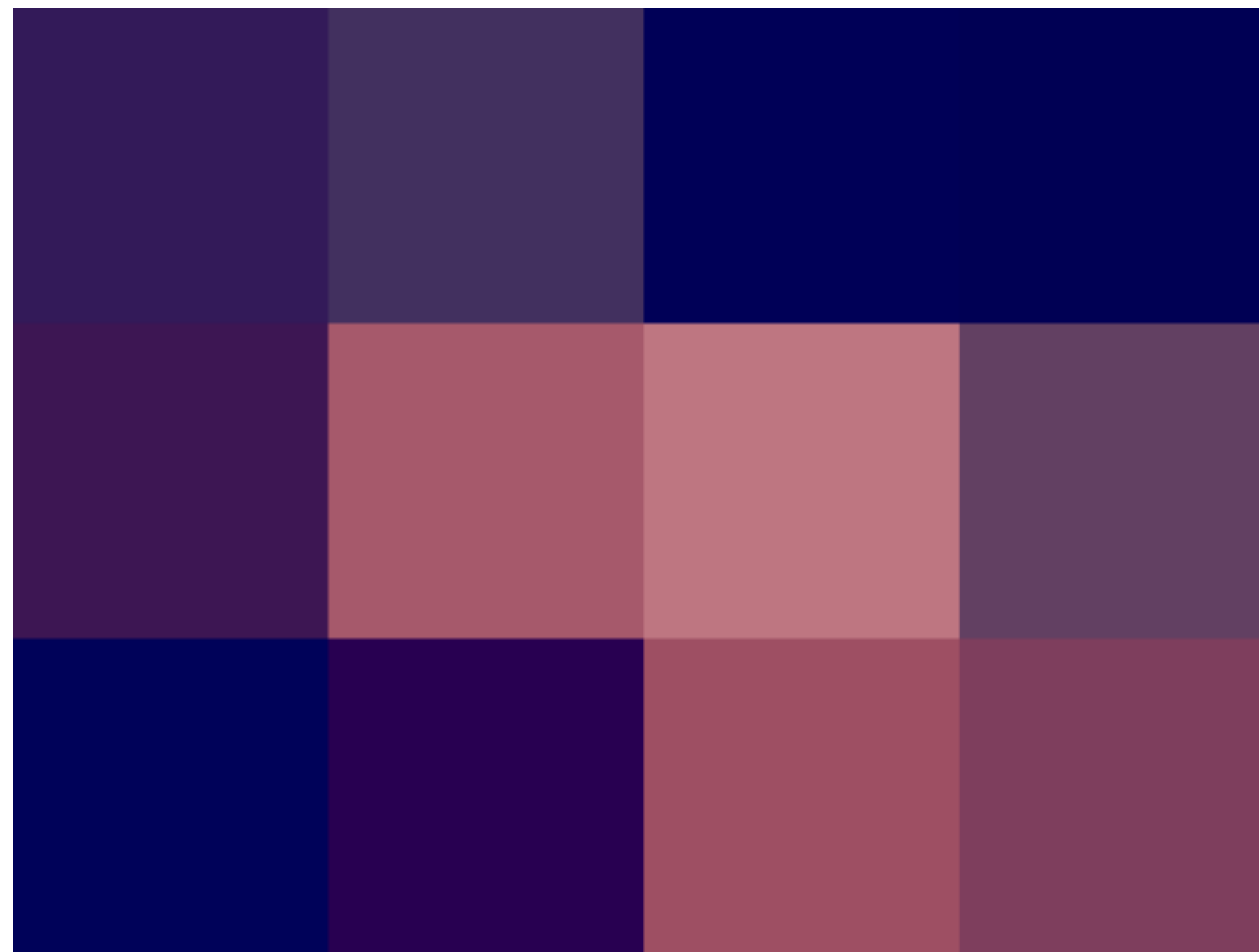


Comment traiter une image?

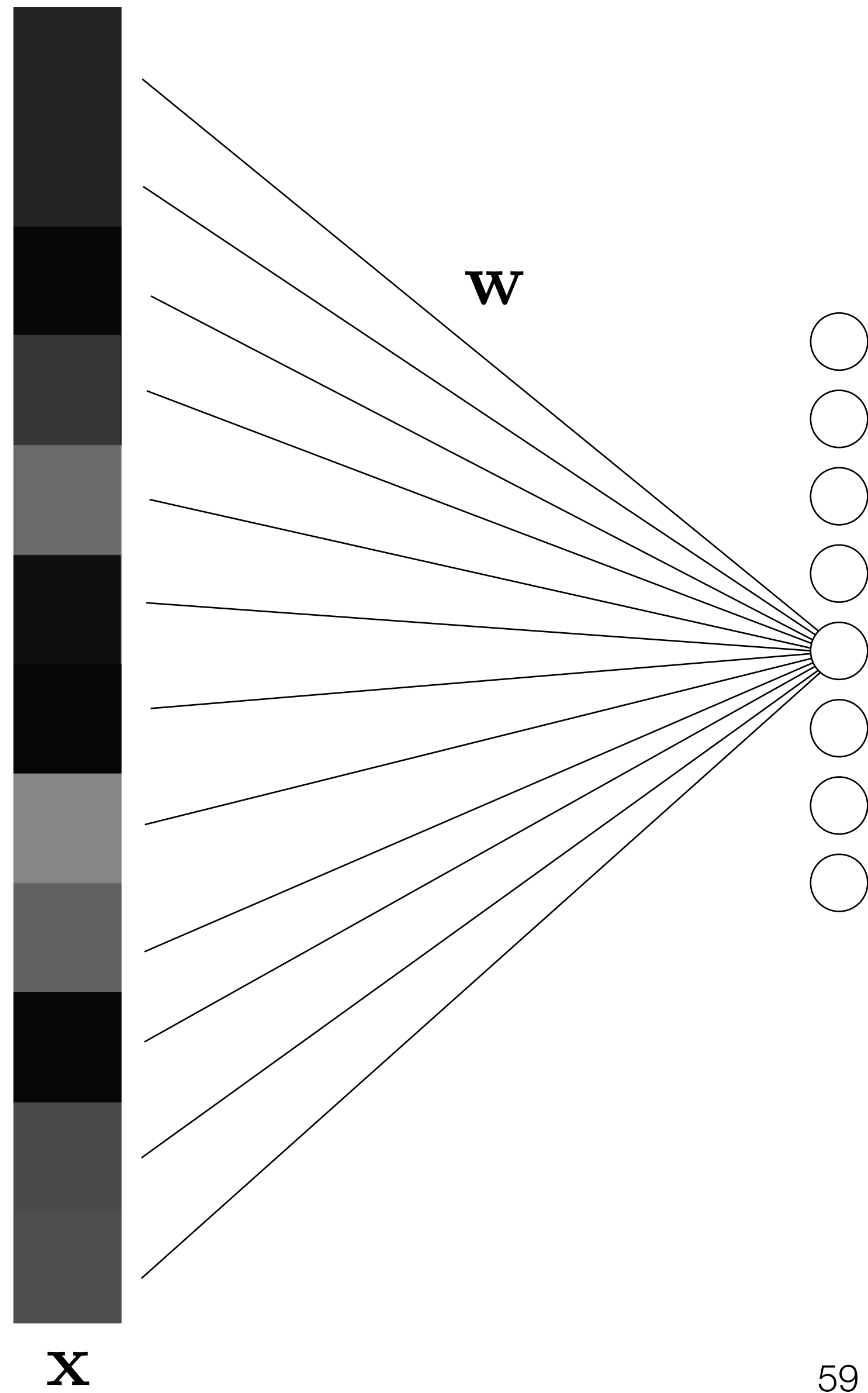


Comment traiter une image?

Ex : résolution 3x4

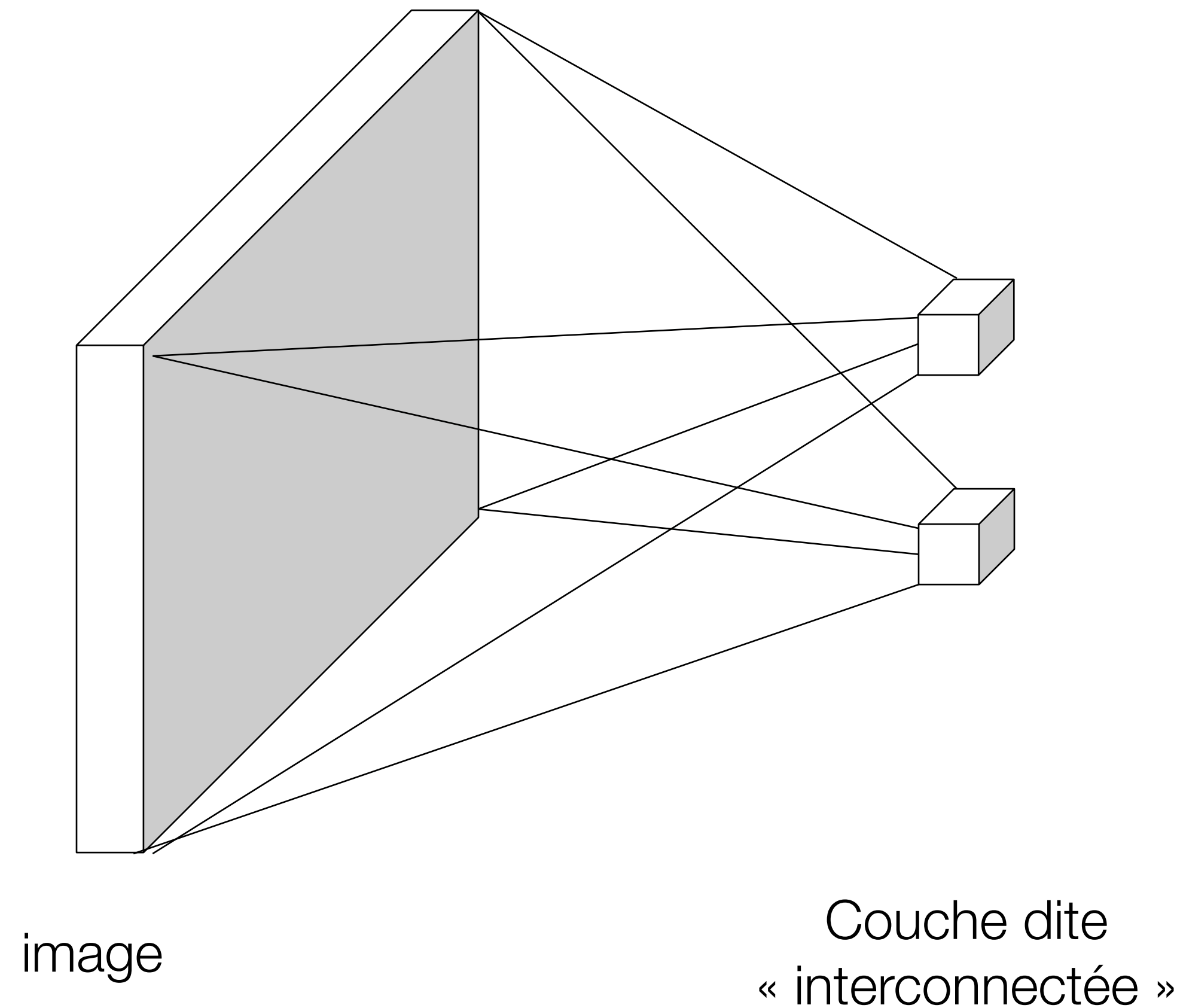


Comment traiter une image?



Réseau de neurone pour image

Quel est le problème?



Réseau de neurone pour image

Puissances de 10

Combien existe-t-il d'images?

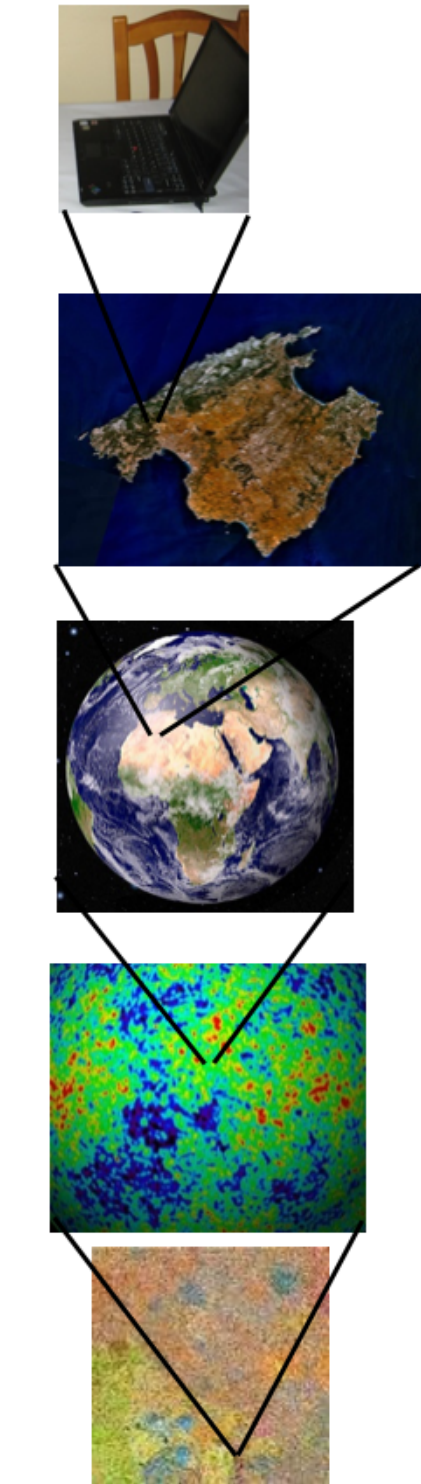
Nombre d'images sur un ordinateur	10^4
Nombre d'images vues durant mes 10 premières années (3 images/seconde * 60 * 60 * 16 * 365 * 10 = 630720000)	10^8
Nombre d'images vues par l'humanité ¹ 108 000 000 000 humains * 60 ans * 3 images/seconde * 60 * 60 * 16 * 365 =	10^{20}
Nombre de photons dans l'univers	10^{88}
Nombre d'images RGB de résolution 32x32 $256^{(32*32*3)}$	10^{7373}

¹ <http://www.prb.org/Articles/2002/HowManyPeopleHaveEverLivedonEarth.aspx>

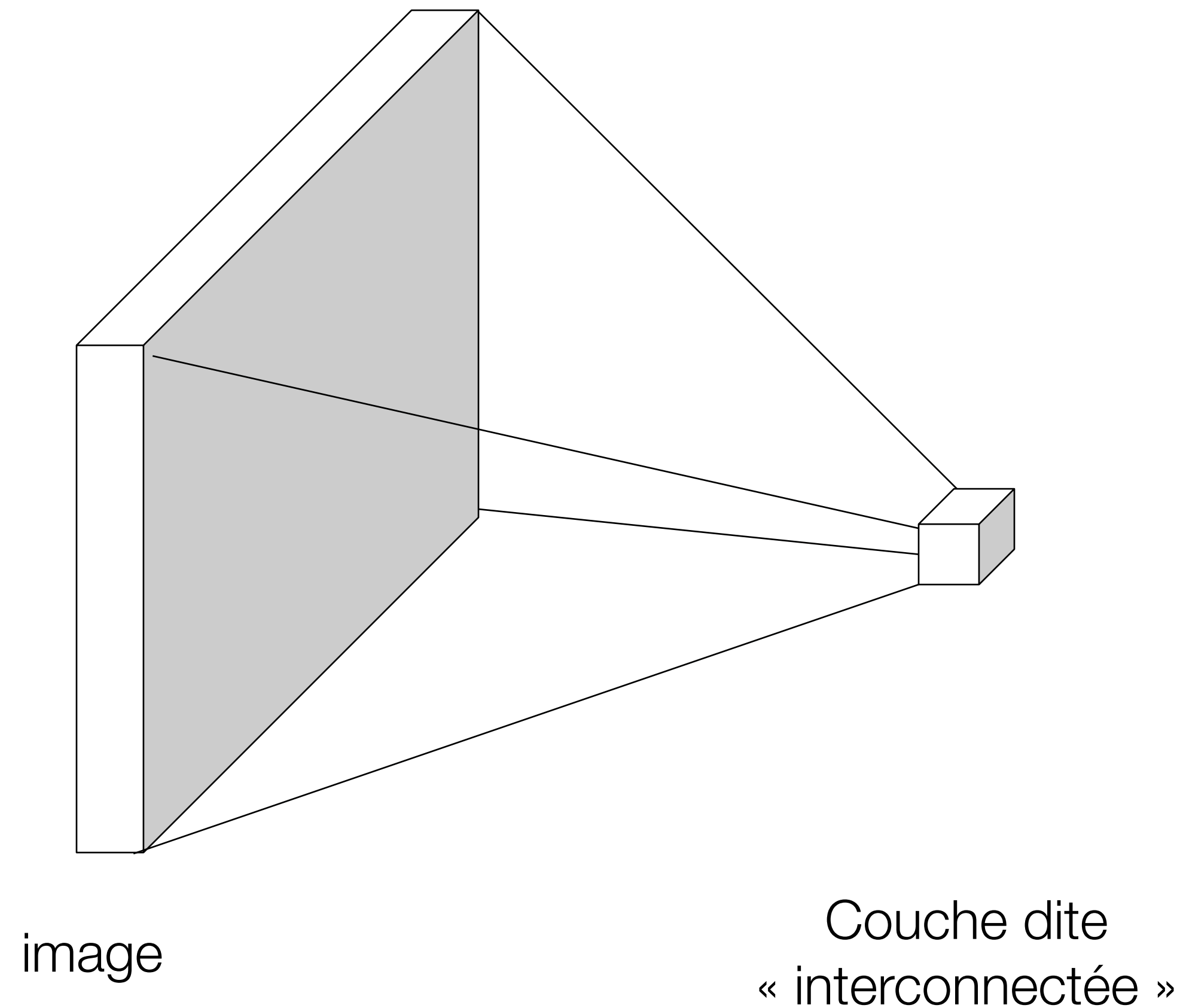
image

Couche dite
« interconnectée »

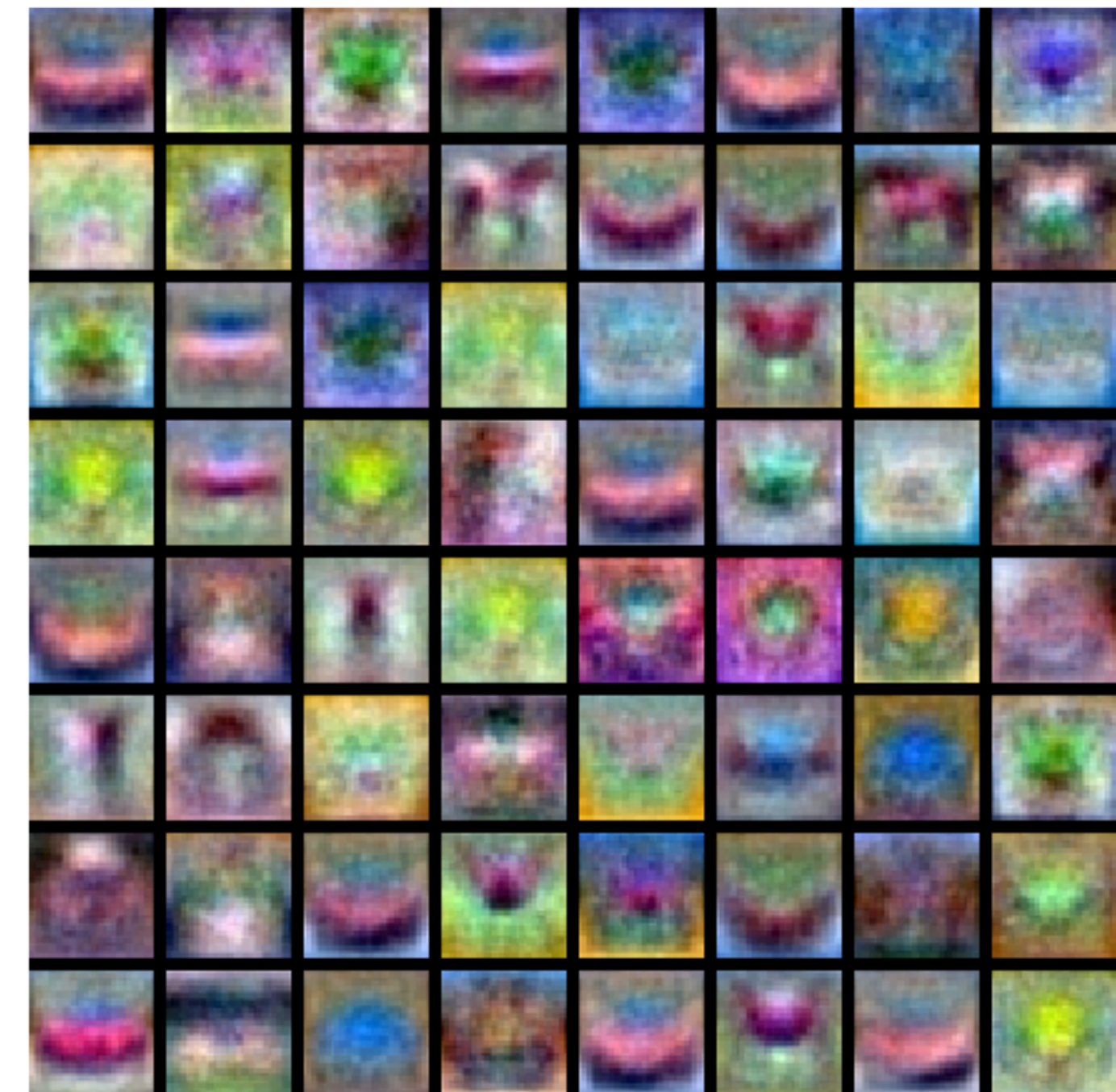
Quel est le problème?



Réseau de neurone pour image

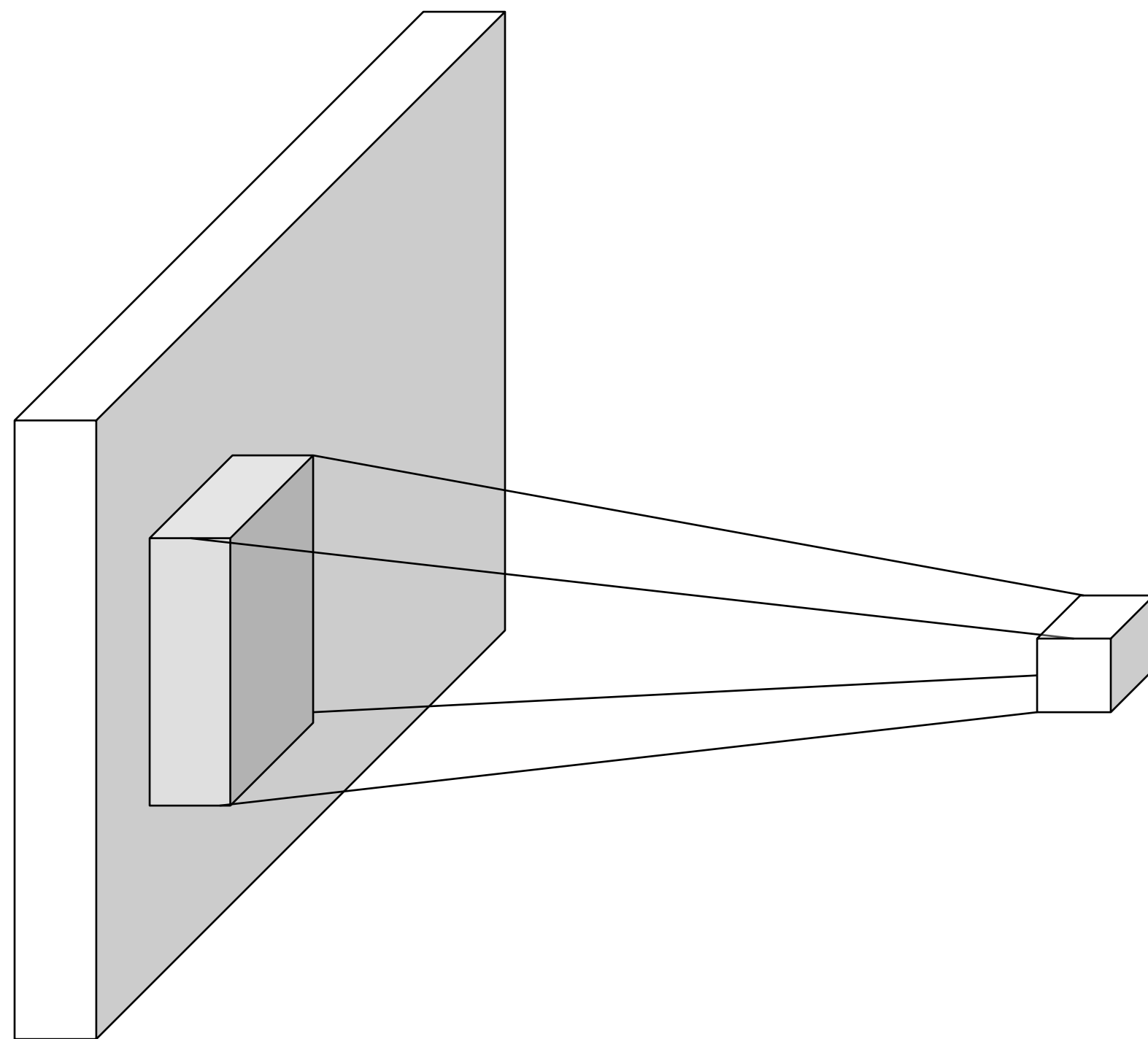


Quel est le problème?



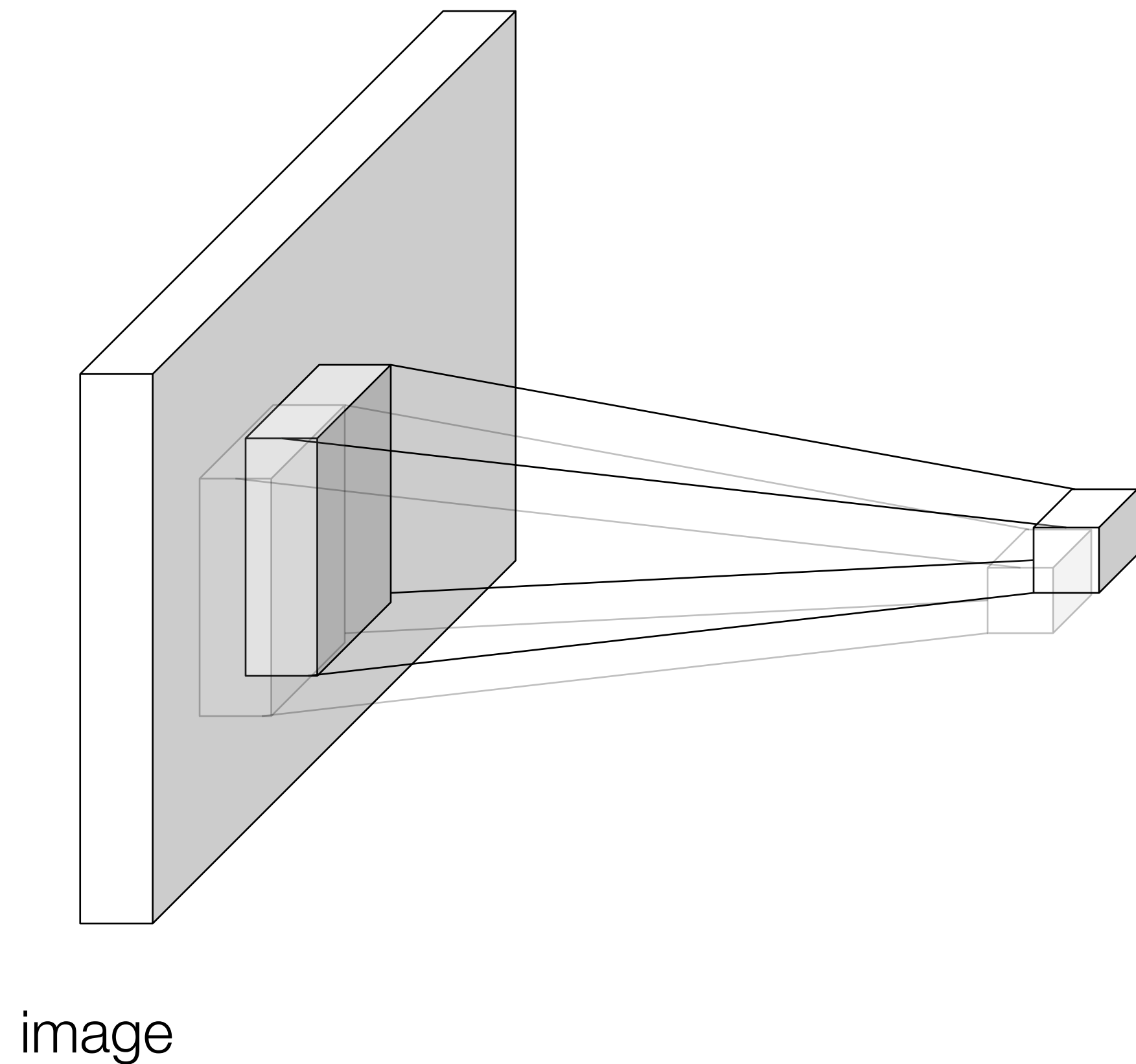
Solution : convolutions!

- Modifications :
 - limiter le champ réceptif de chaque neurone
 - partager les poids



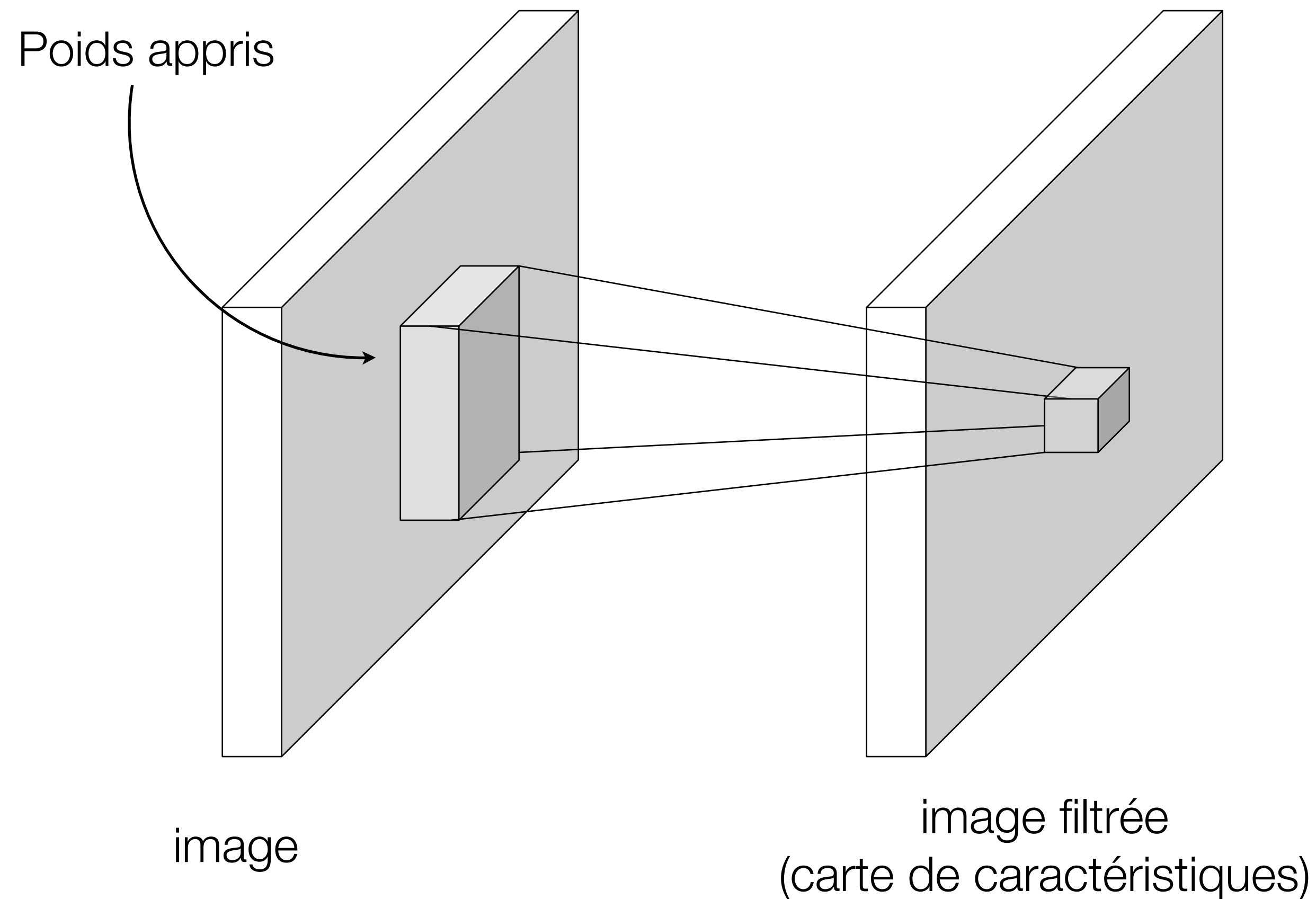
image

Solution : convolutions!



- Modifications :
 - limiter le champ réceptif de chaque neurone
 - partager les poids
 - répéter l'action sur toute l'image

Solution : convolutions!

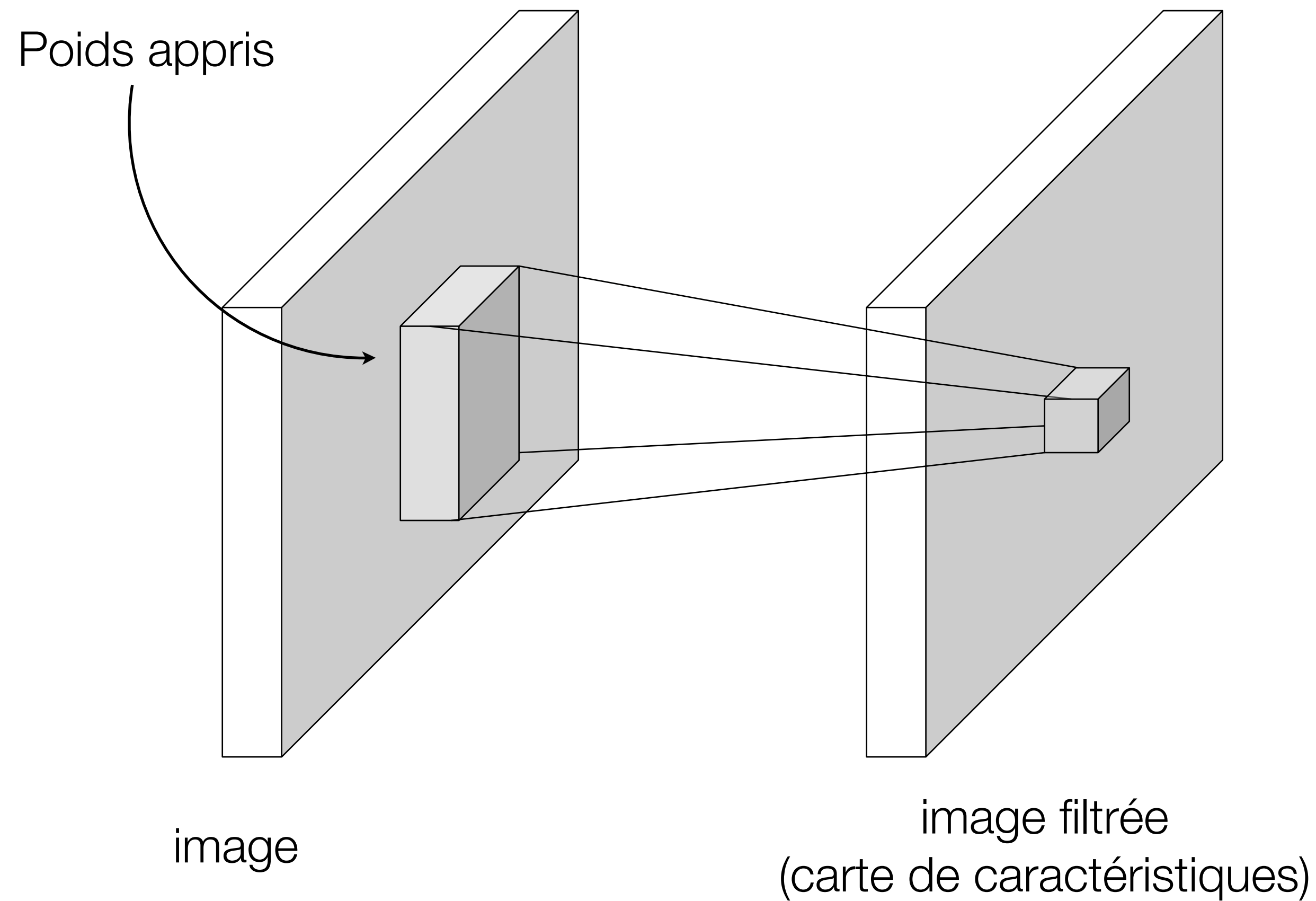


- Modifications :
- limiter le champ réceptif de chaque neurone
- partager les poids
- répéter l'action sur toute l'image

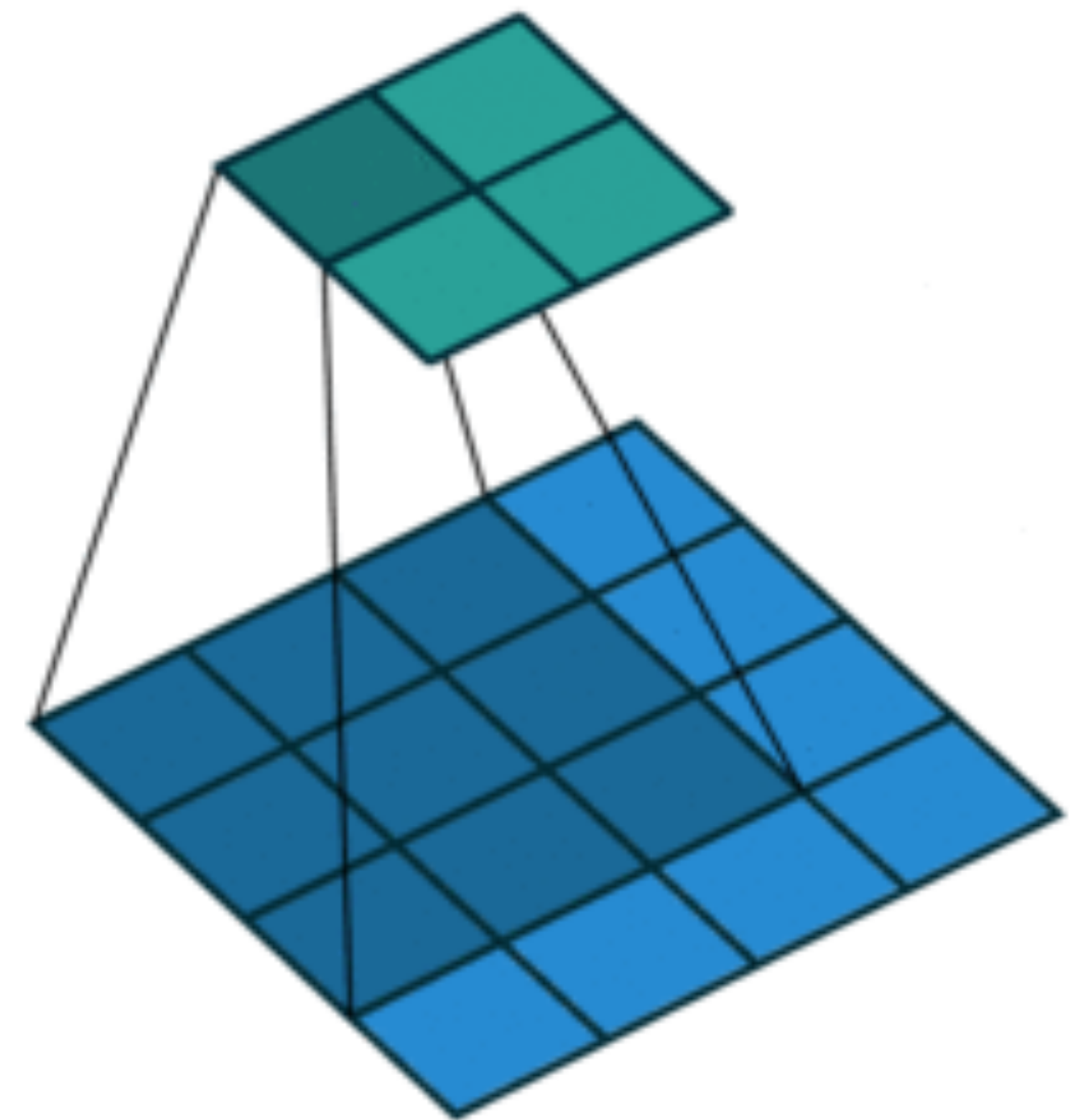
Ceci équivaut à calculer
la convolution d'un filtre sur une image!

Ce sont les **poids du filtre**
qui doivent être appris.

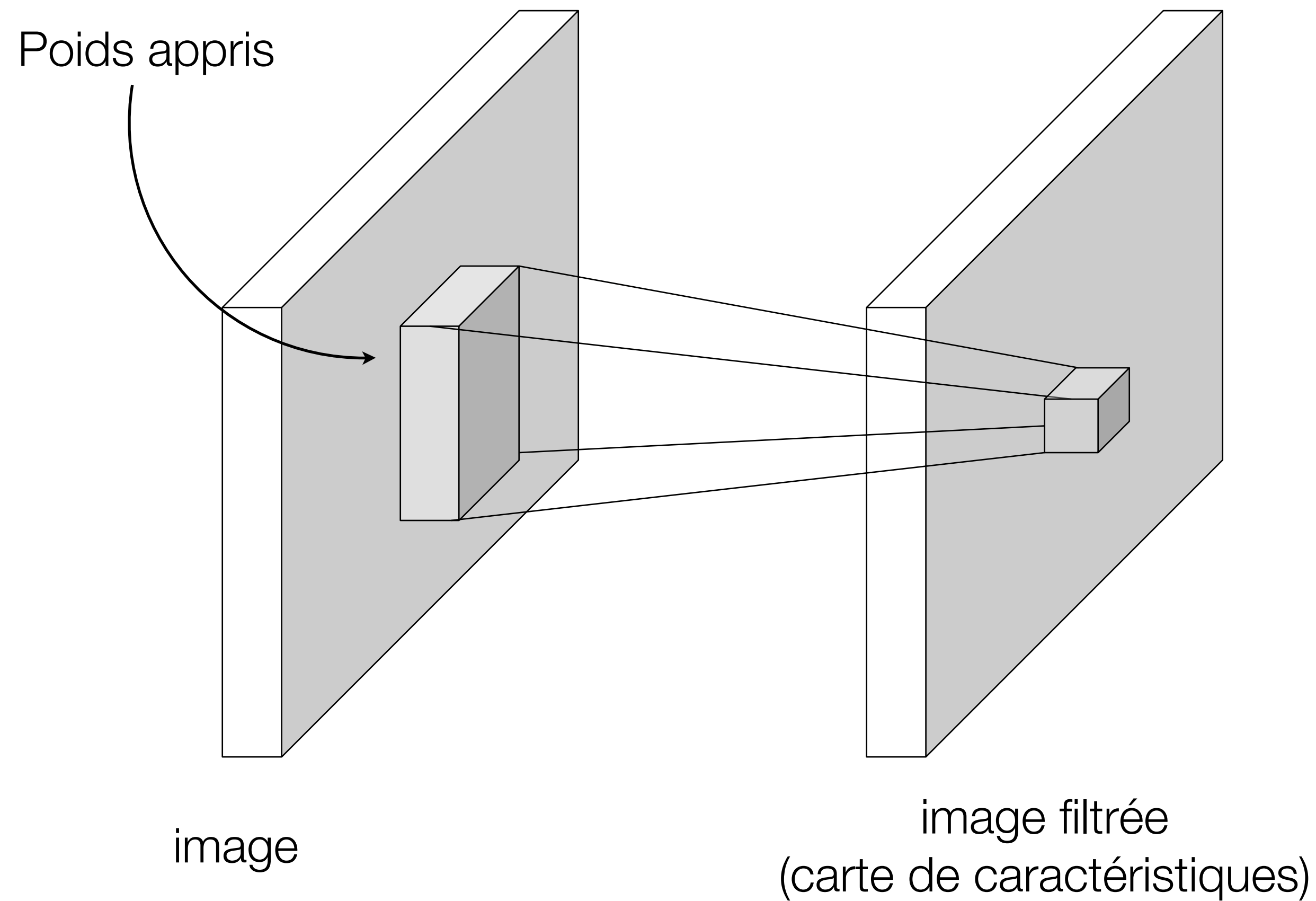
Solution : convolutions!



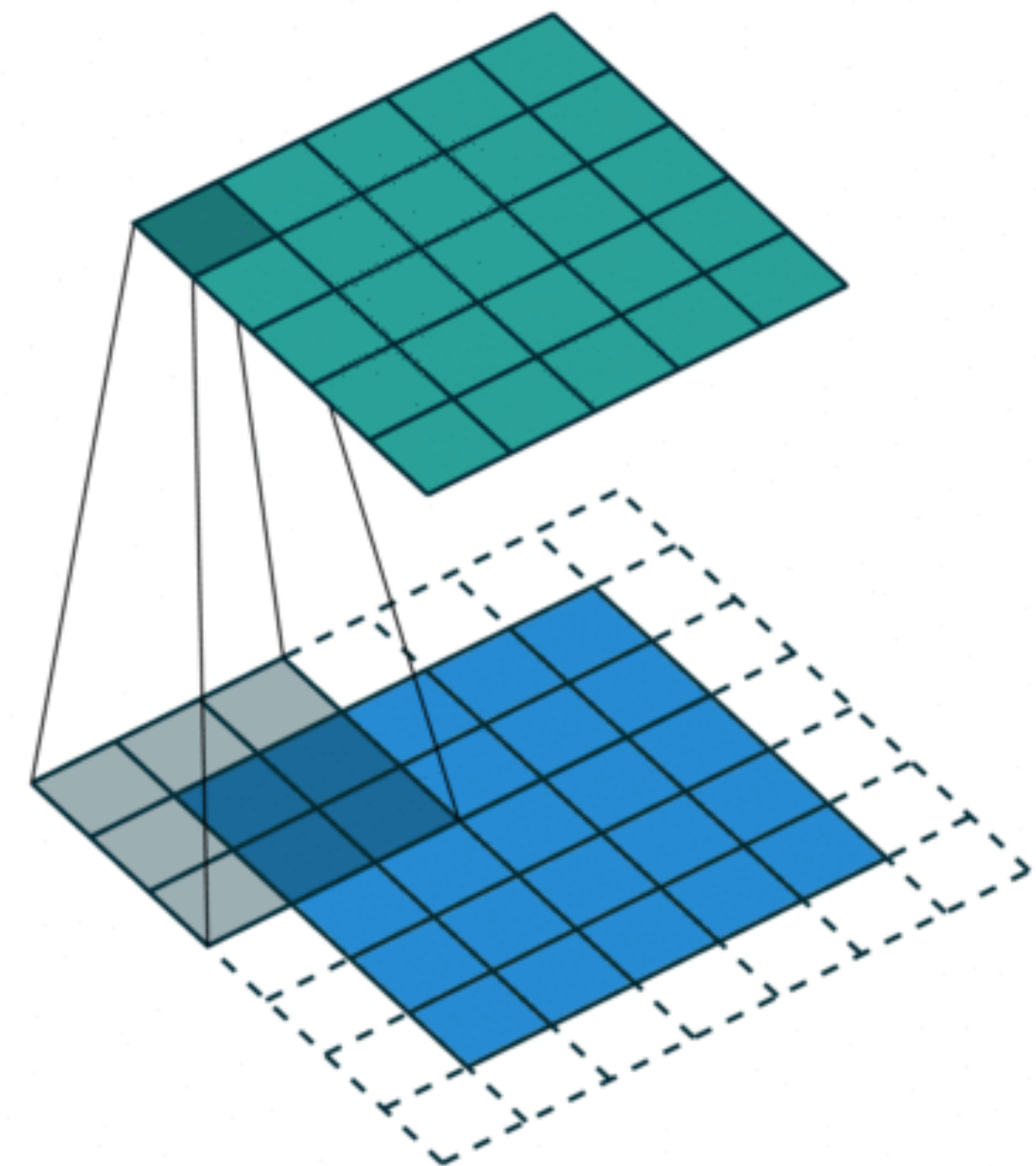
La résolution de l'image filtrée
(caractéristiques) est réduite...



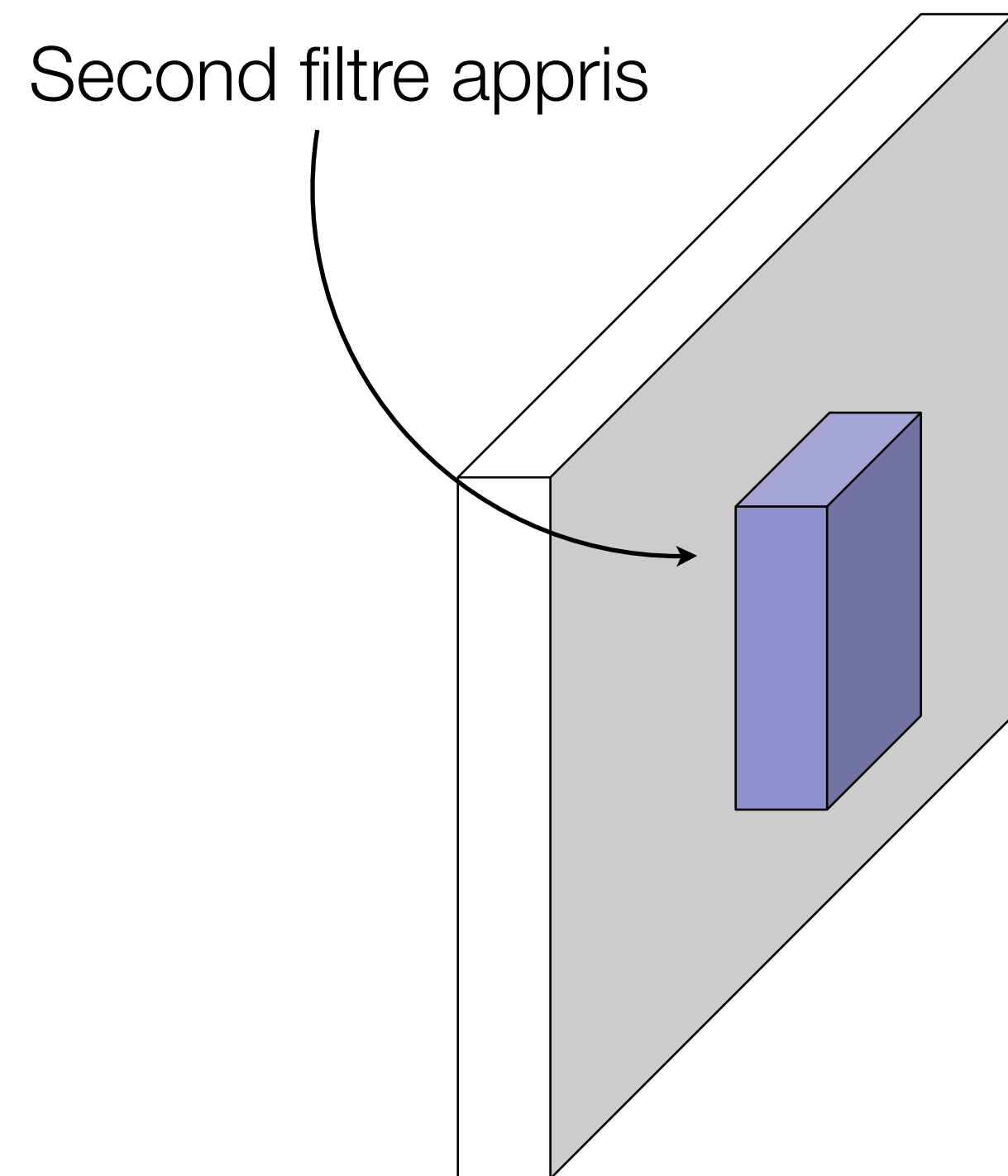
Solution : convolutions!



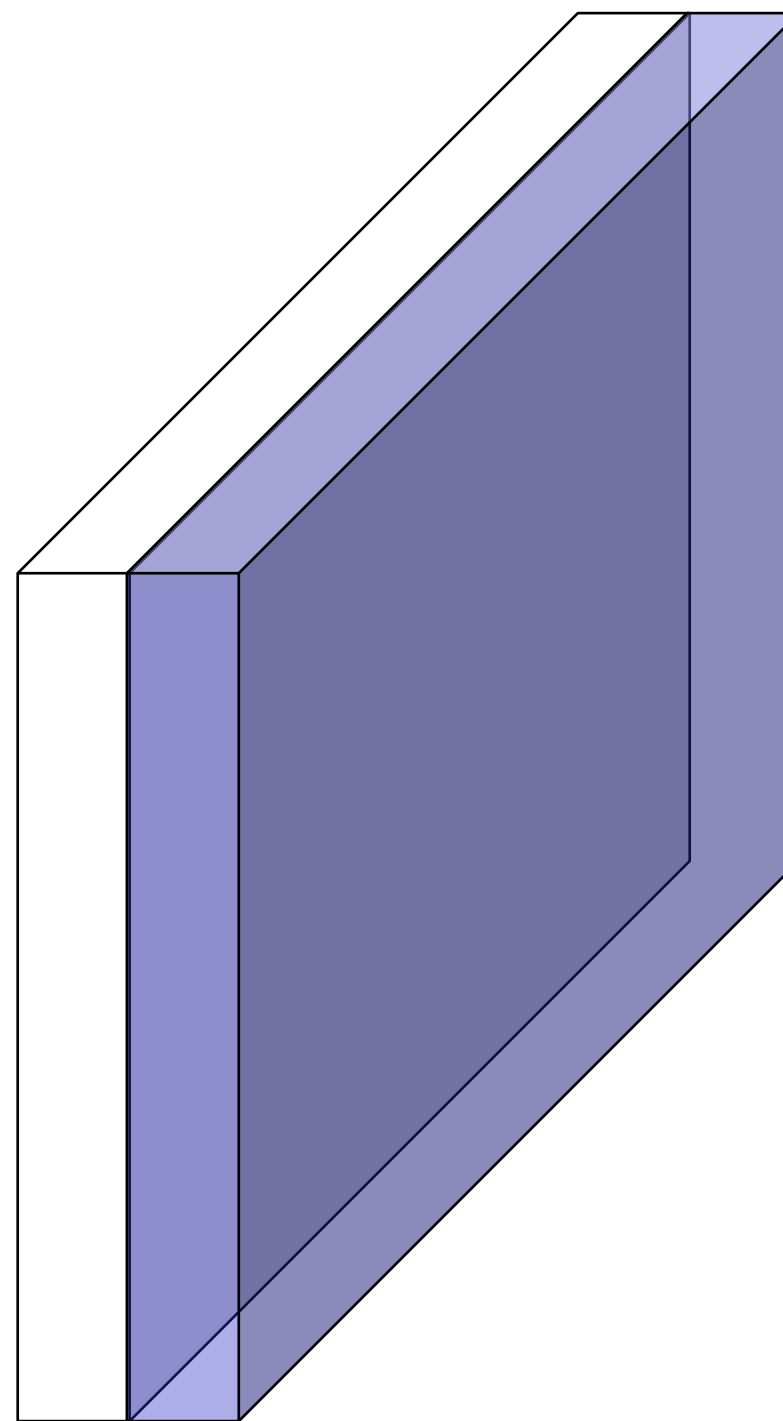
Solution :
on rajoute des 0 (*0-padding*)



Anatomie d'une couche de convolution



image



Seconde carte de caractéristiques

Anatomie d'une couche de convolution

Rappel : banques de filtres

Second filtre appris

Comment peut-on la représenter?

- Calculer les caractéristiques des arêtes à différentes orientations, et à différentes échelles

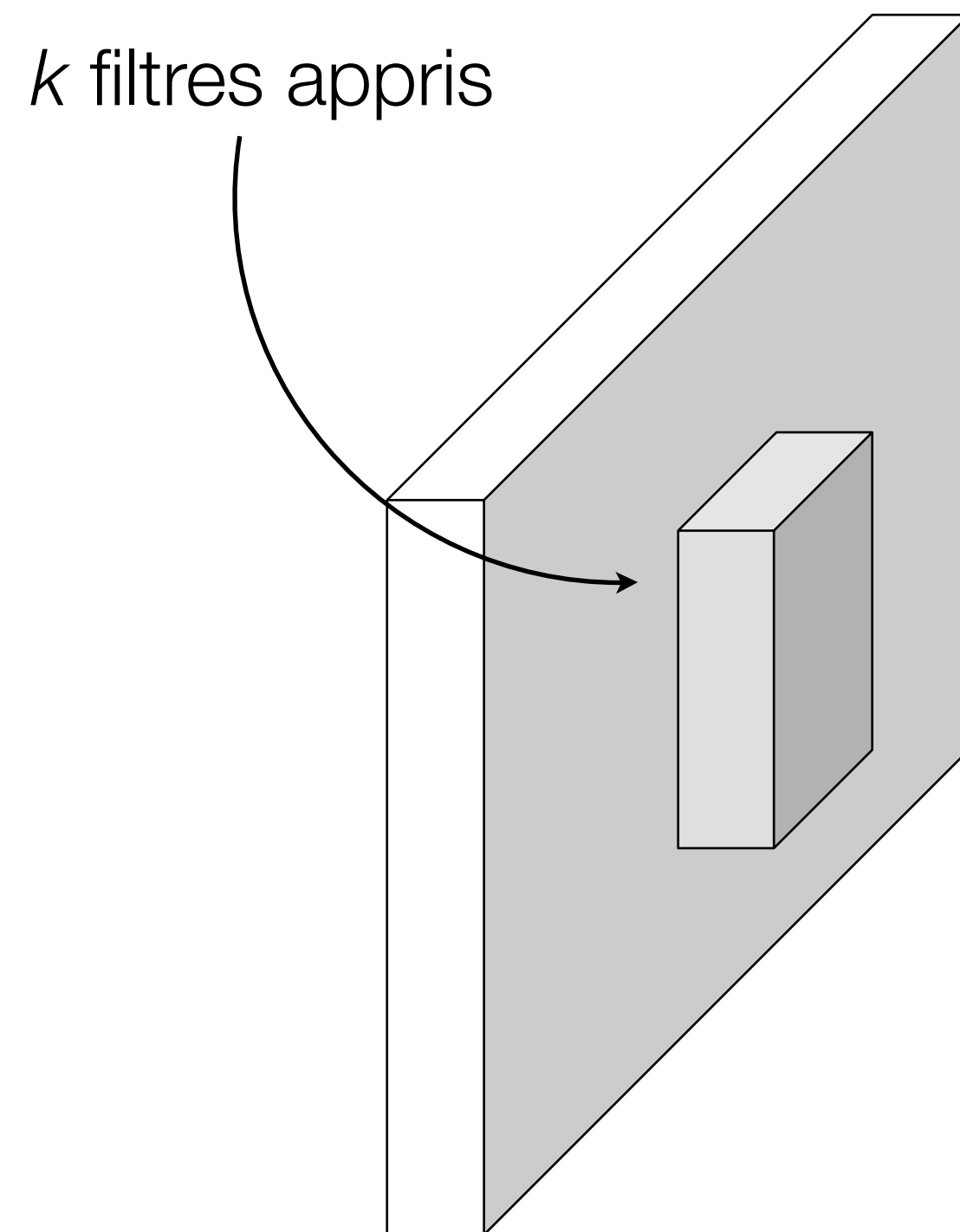
Plutôt que d'utiliser une banque de filtres prédéterminée, les poids de chaque filtre sont **appris** à partir des données!

image

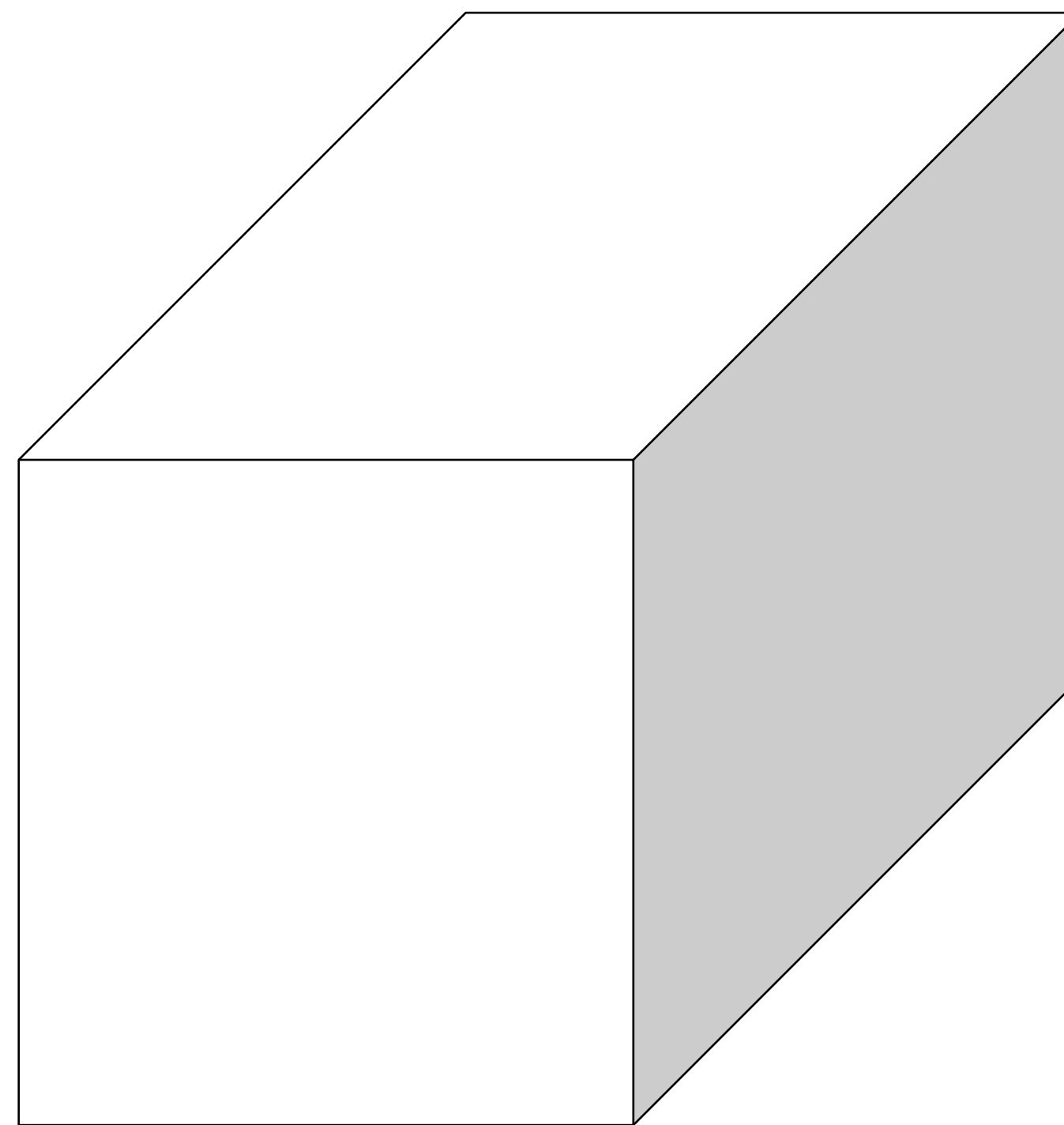
T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. International Journal of Computer Vision, 43(1), 2001

Anatomie d'une couche de convolution

La résolution spatiale est préservée

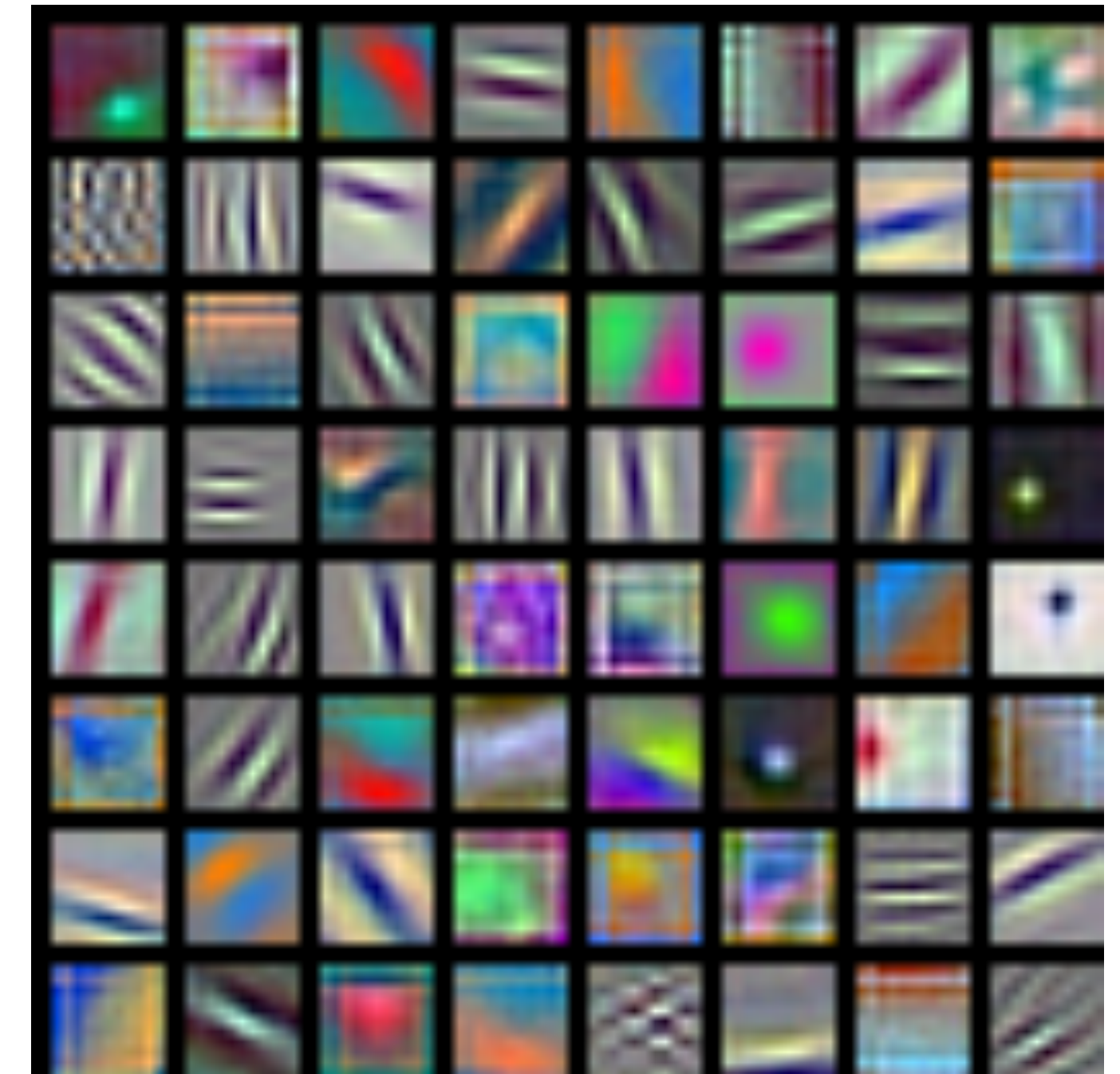


image

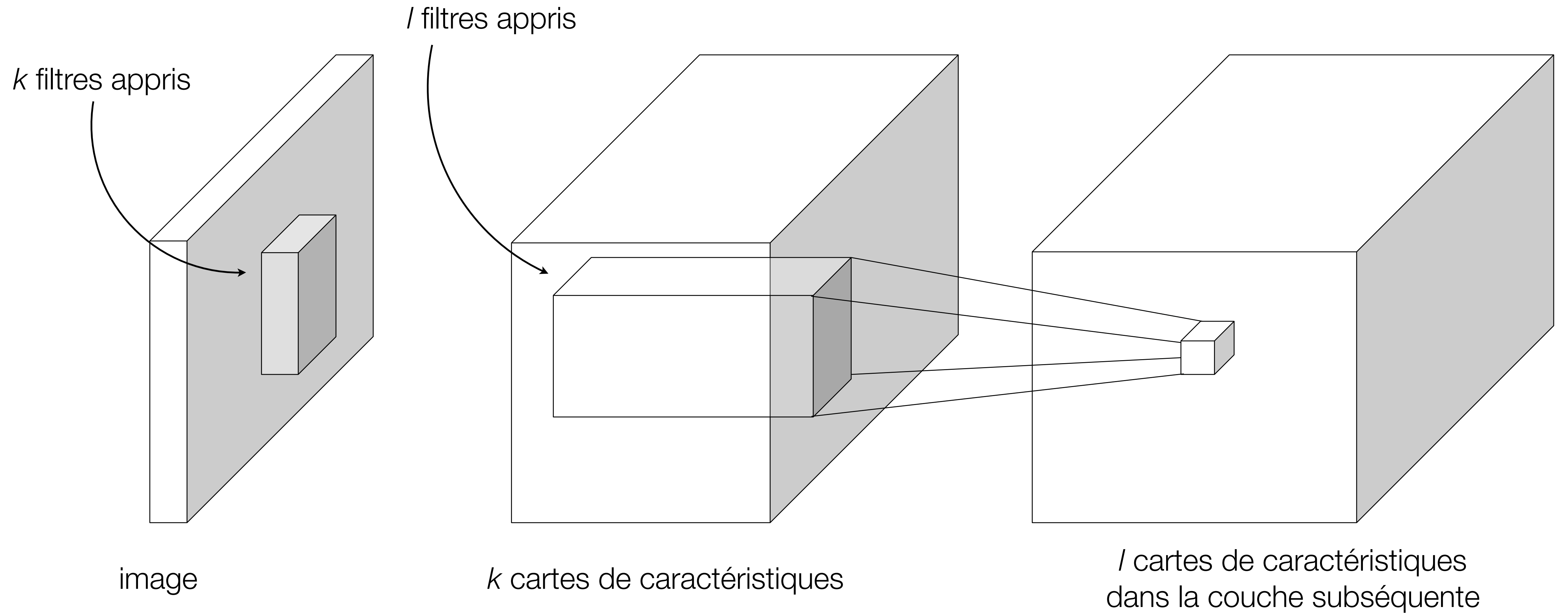


k cartes de caractéristiques

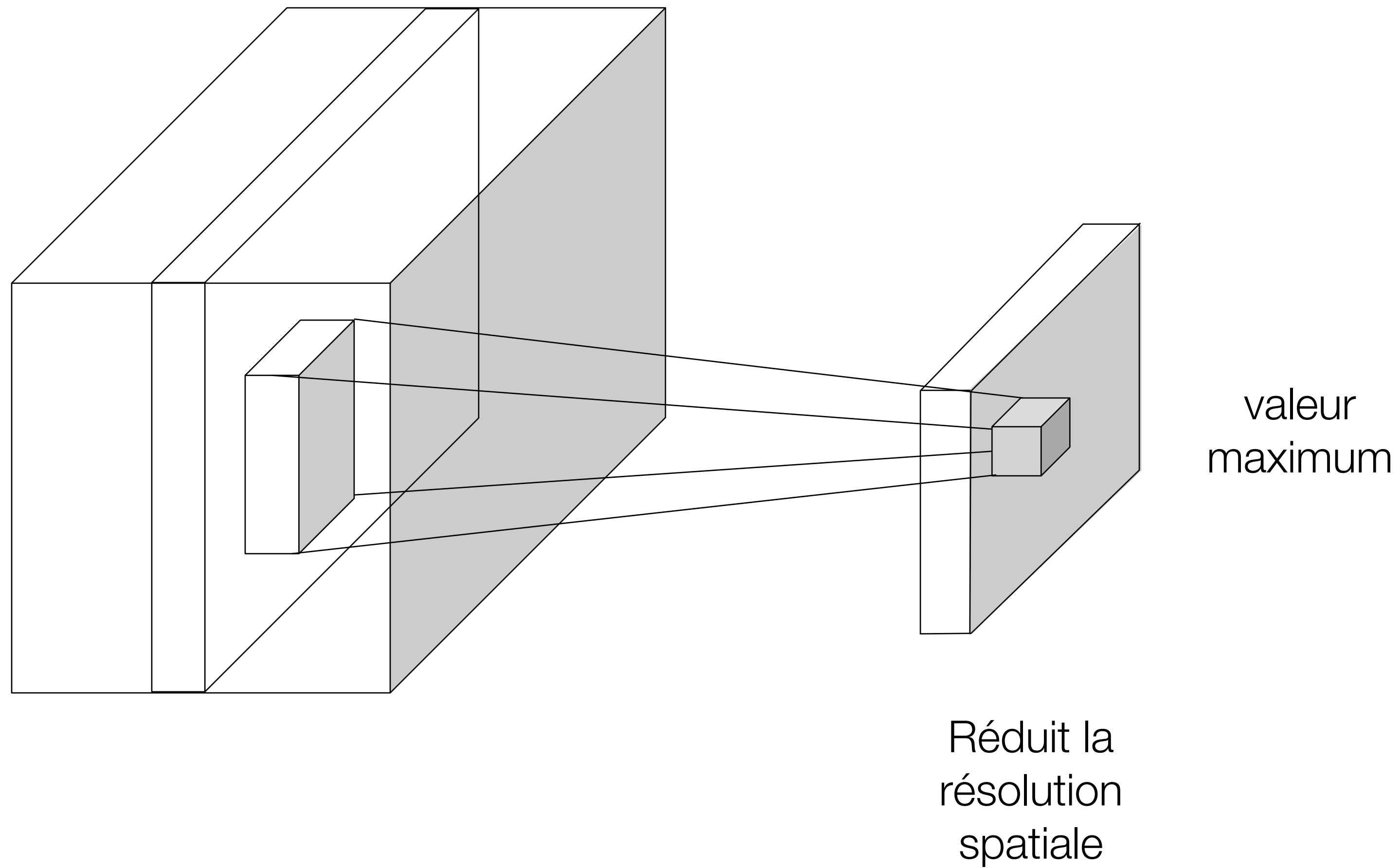
exemples de filtres appris



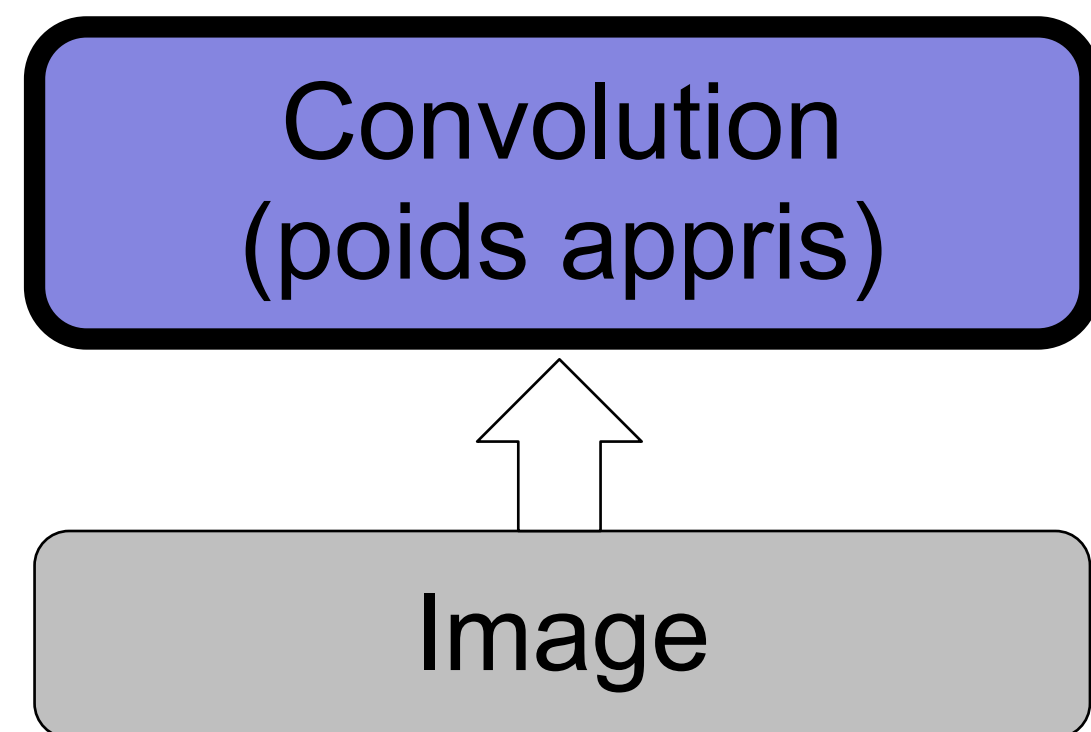
Anatomie d'une couche de convolution



Couche d'agrégation (*pooling*)



L'architecture d'un réseau à convolutions



Image

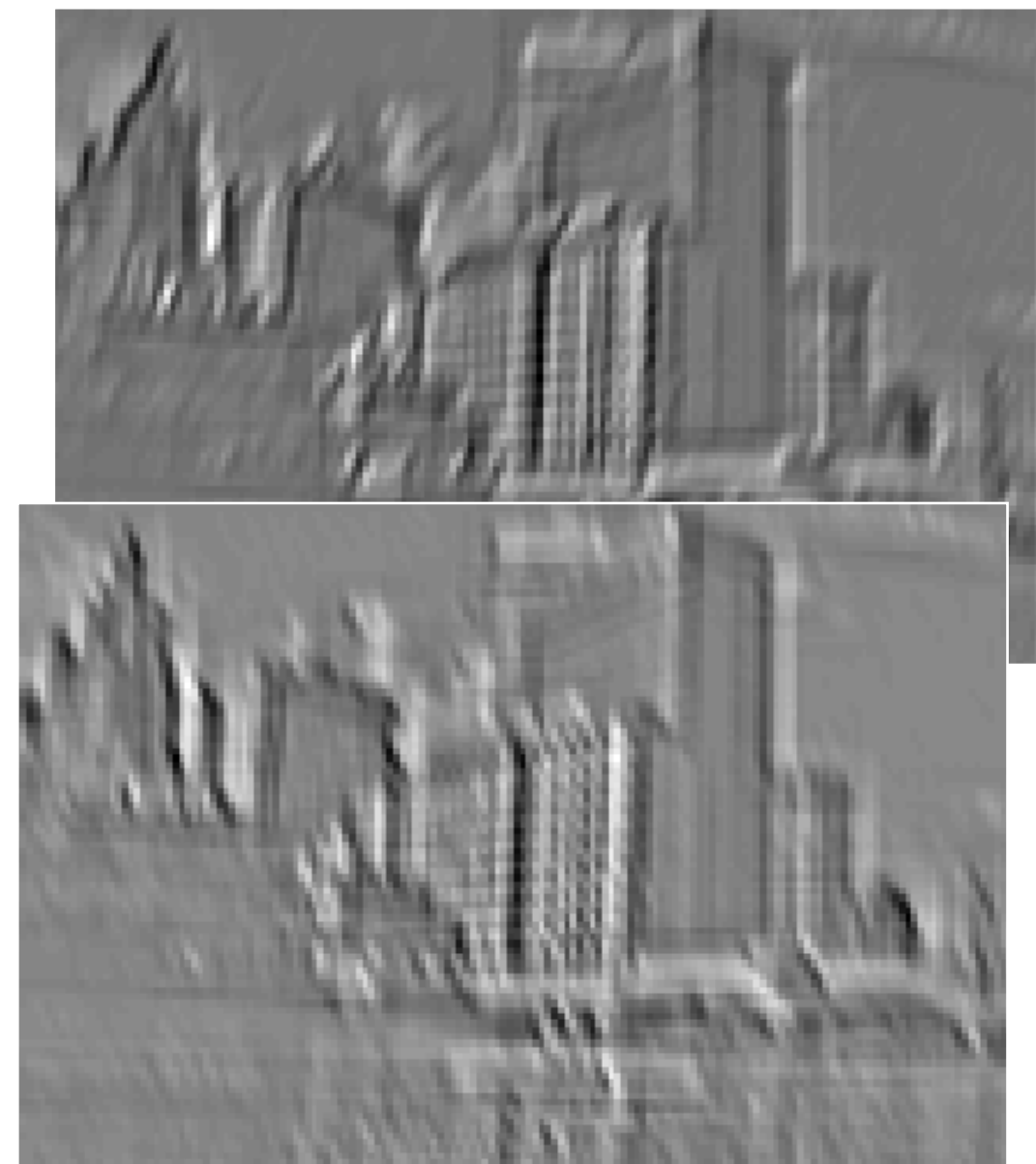
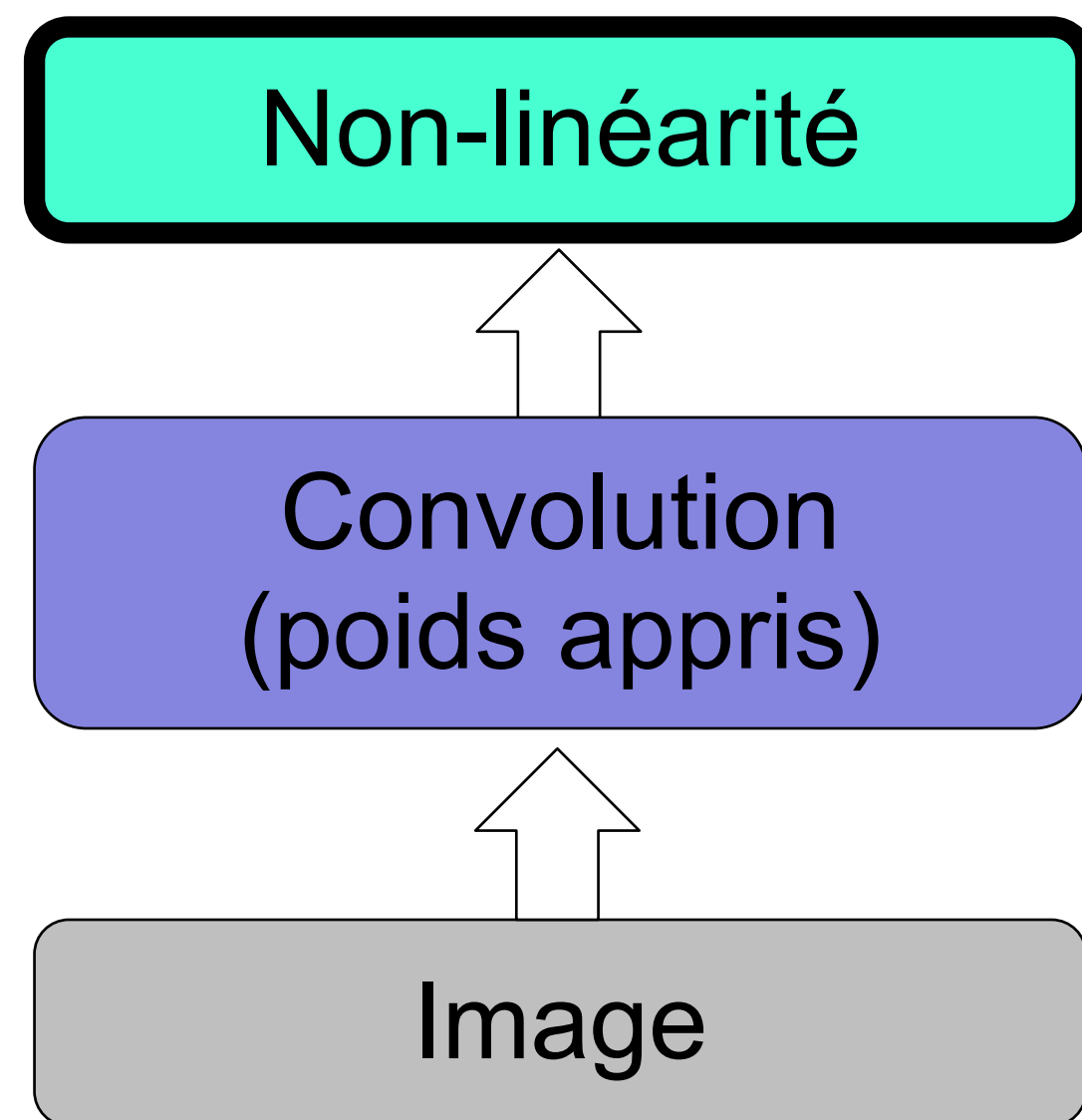
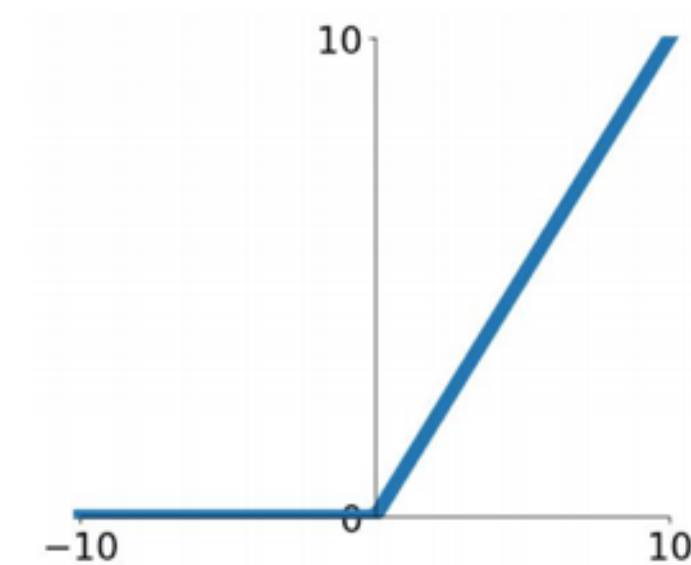


Image convoluée

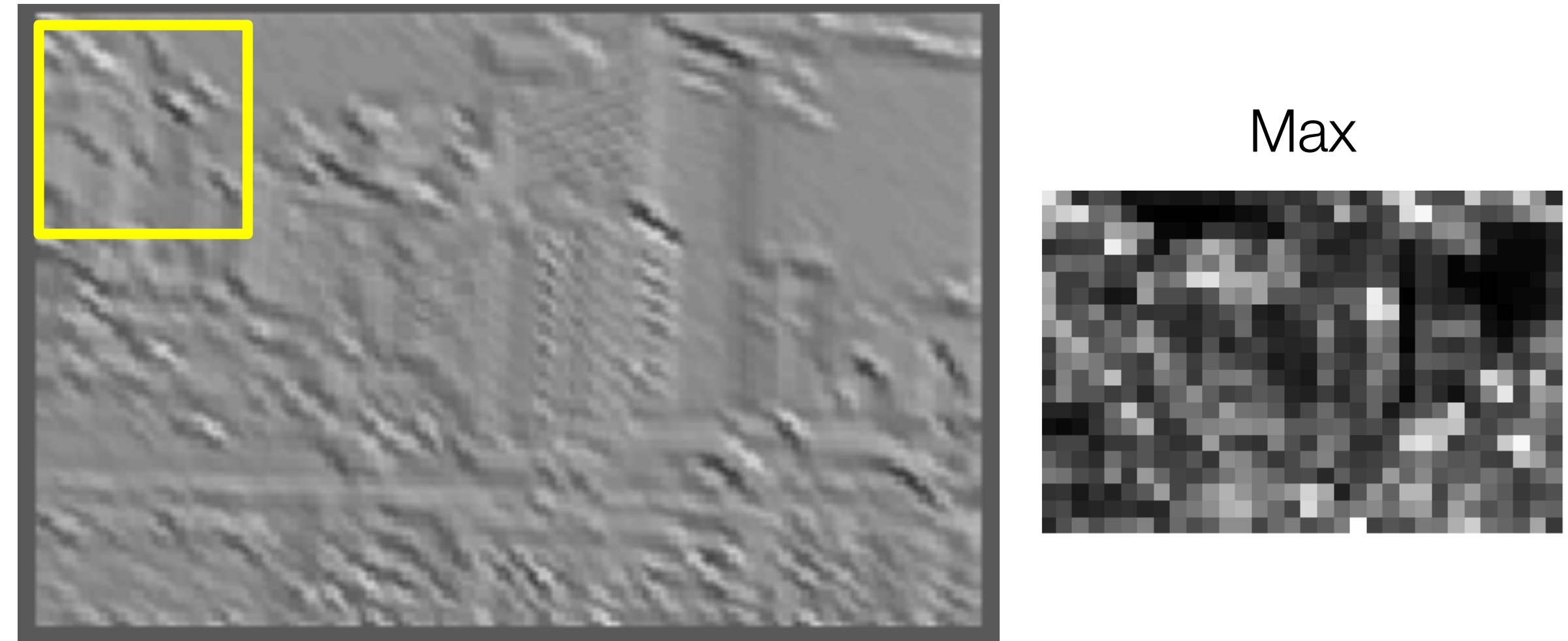
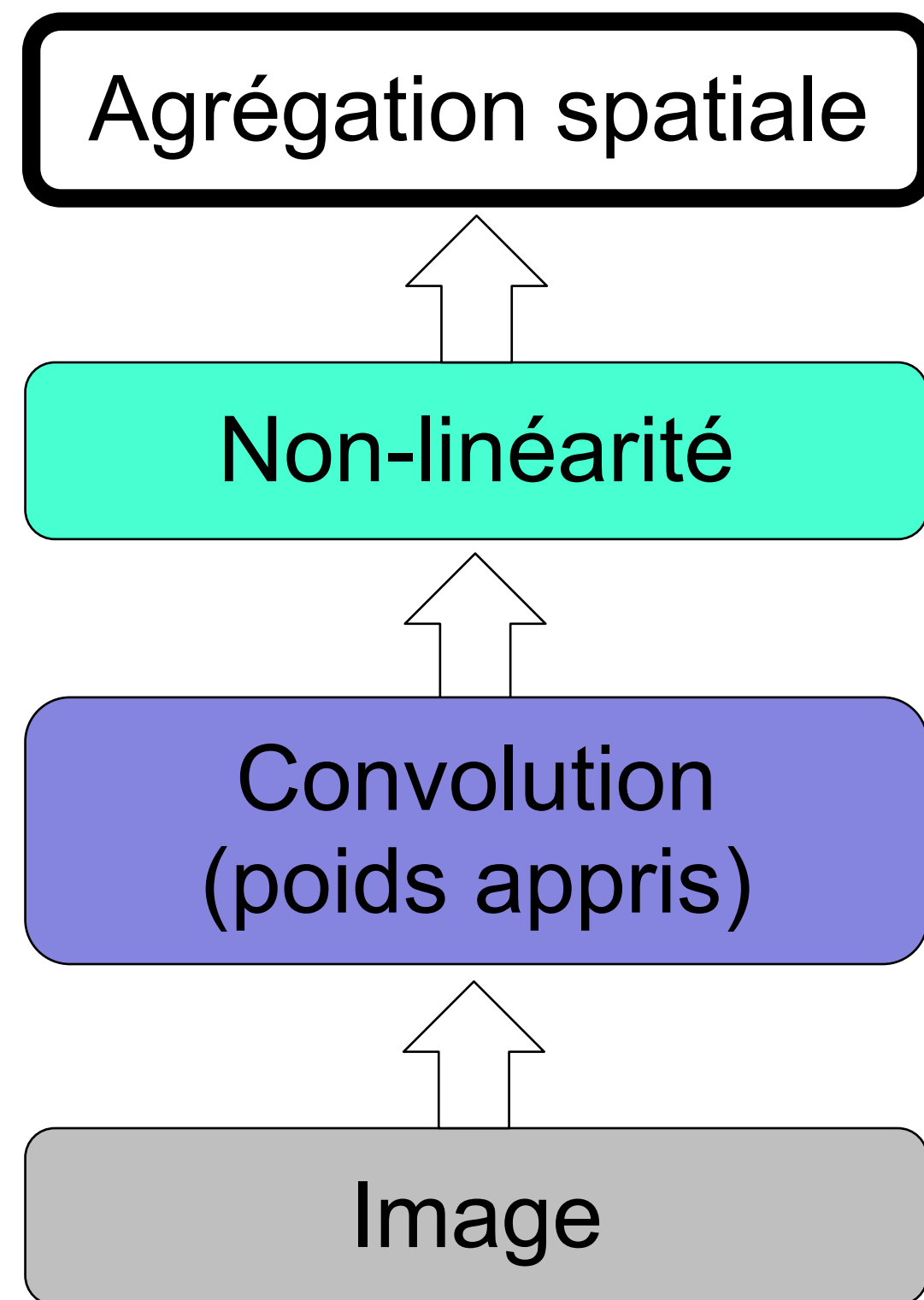
L'architecture d'un réseau à convolutions



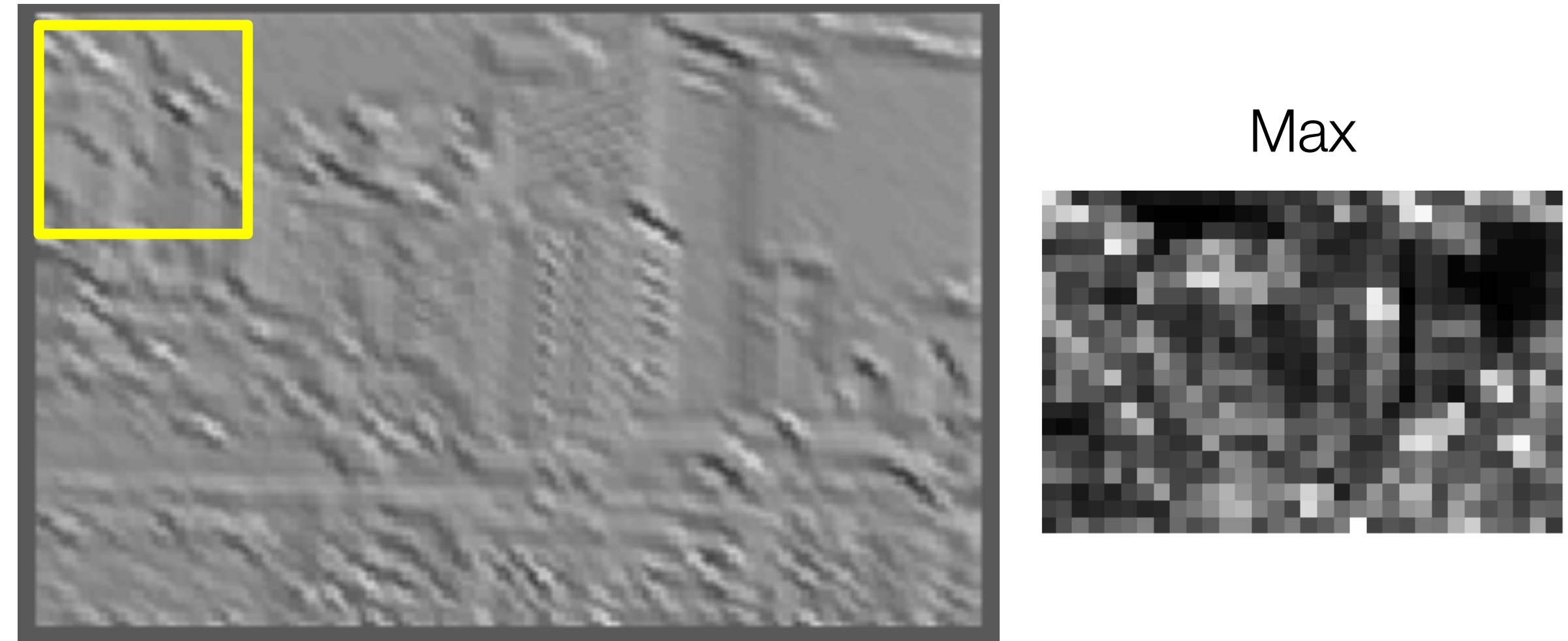
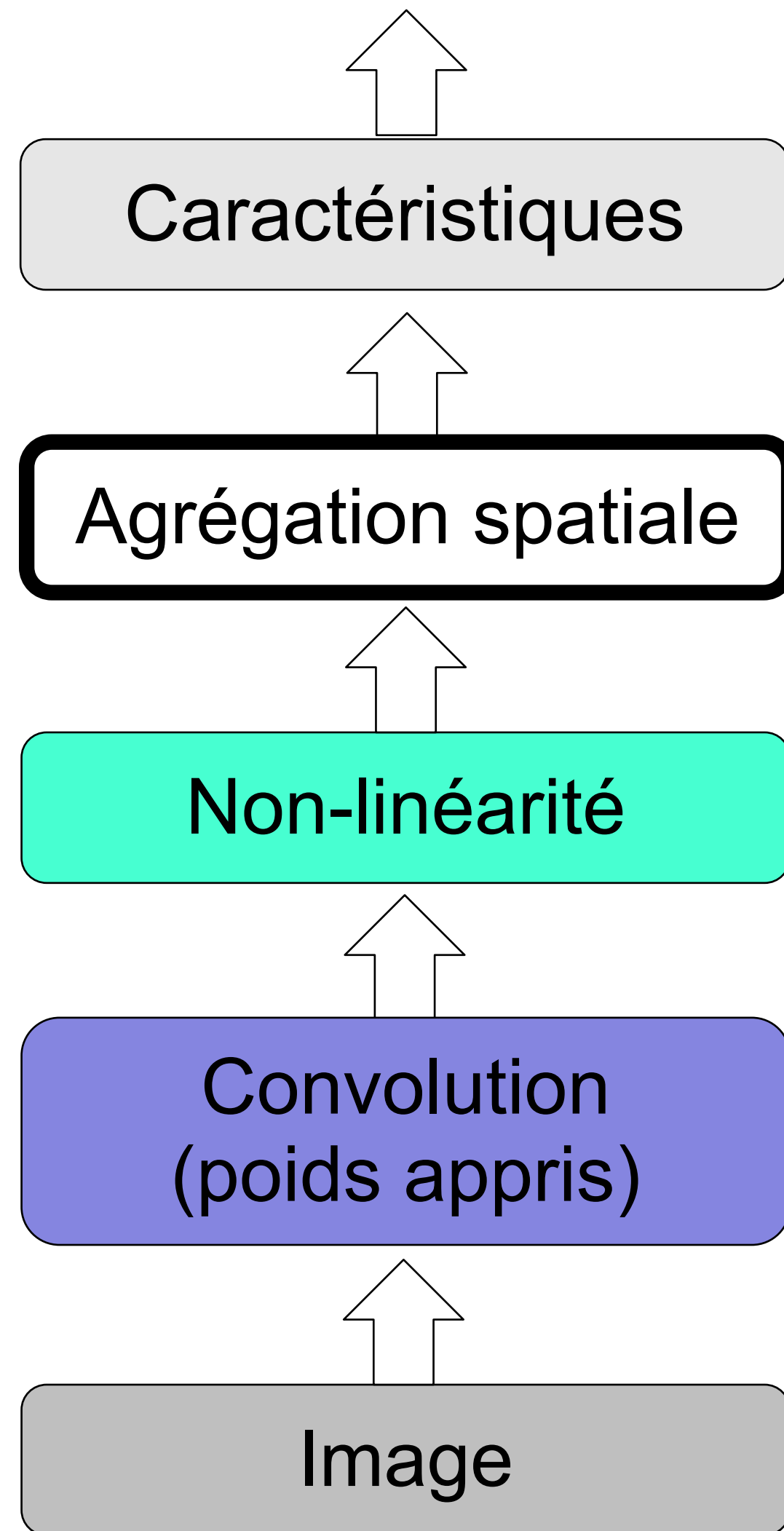
ReLU
 $\max(0, x)$



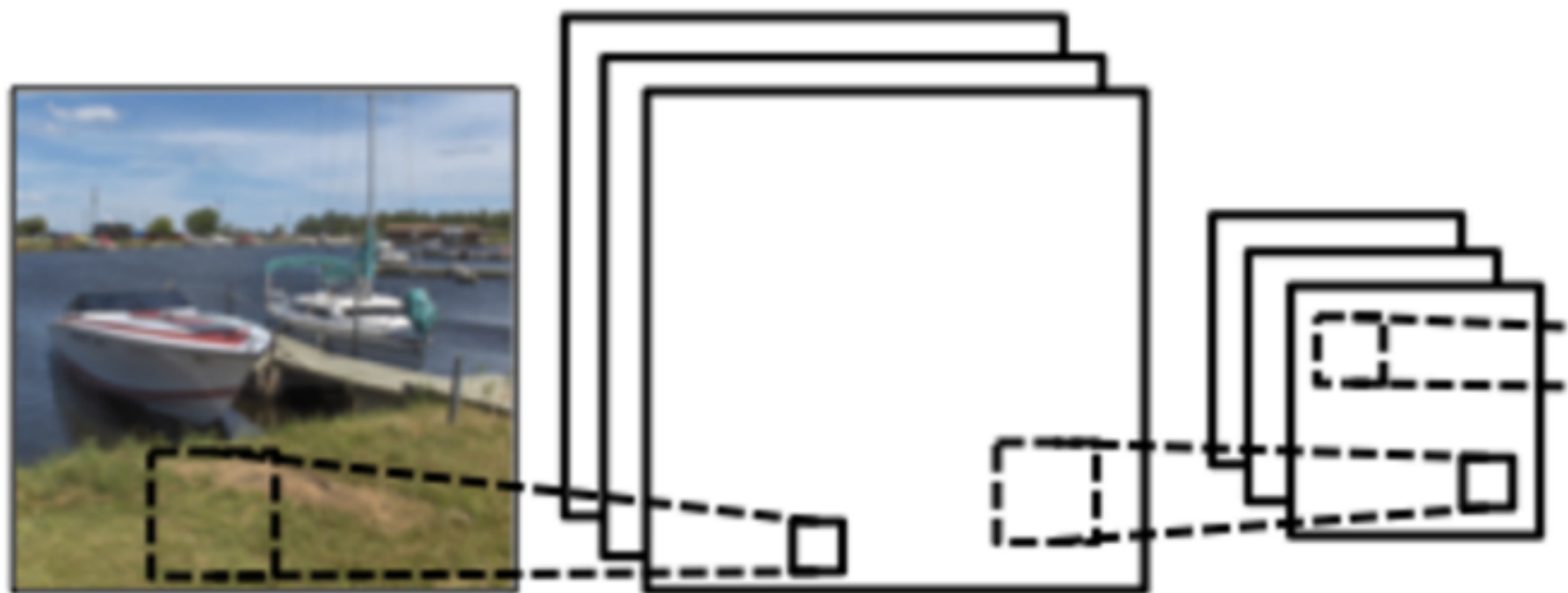
L'architecture d'un réseau à convolutions



L'architecture d'un réseau à convolutions



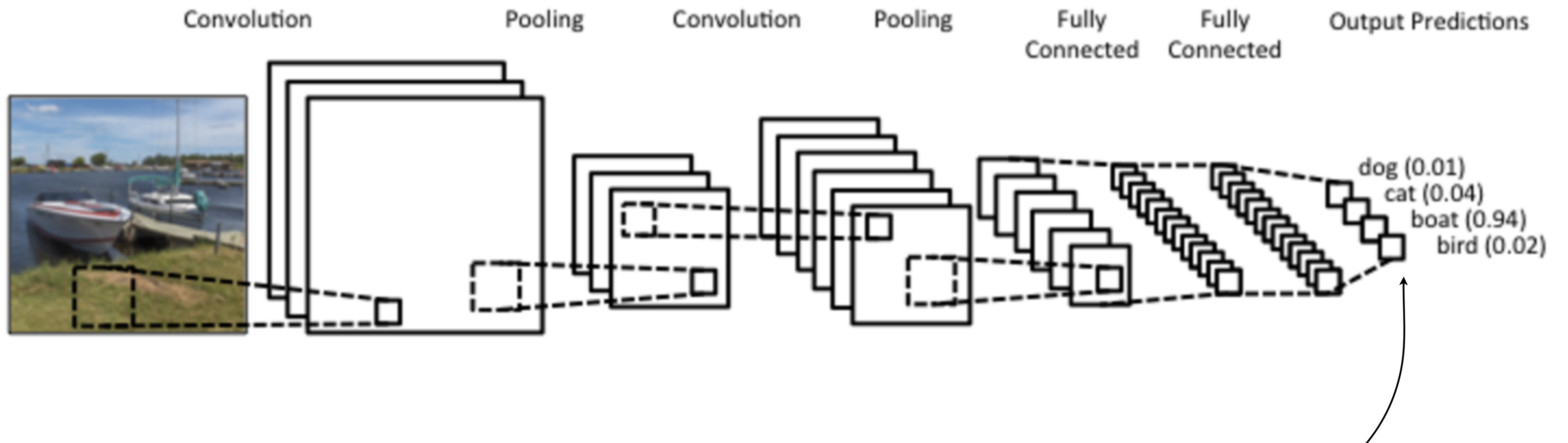
L'architecture d'un réseau à convolutions



L'architecture d'un réseau à convolutions



L'architecture d'un réseau à convolutions

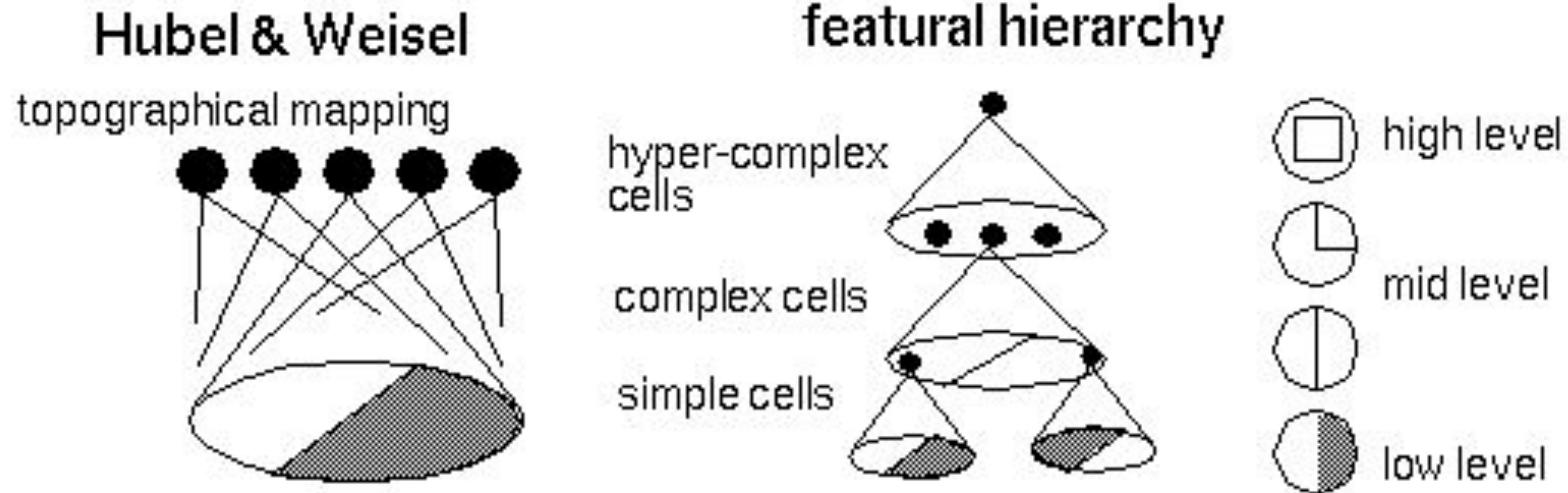


Couche de type « softmax »

$$\text{softmax}(\mathbf{x}_i) = \frac{e^{\mathbf{x}_i}}{\sum_{j=1}^n e^{\mathbf{x}_j}}$$

Source

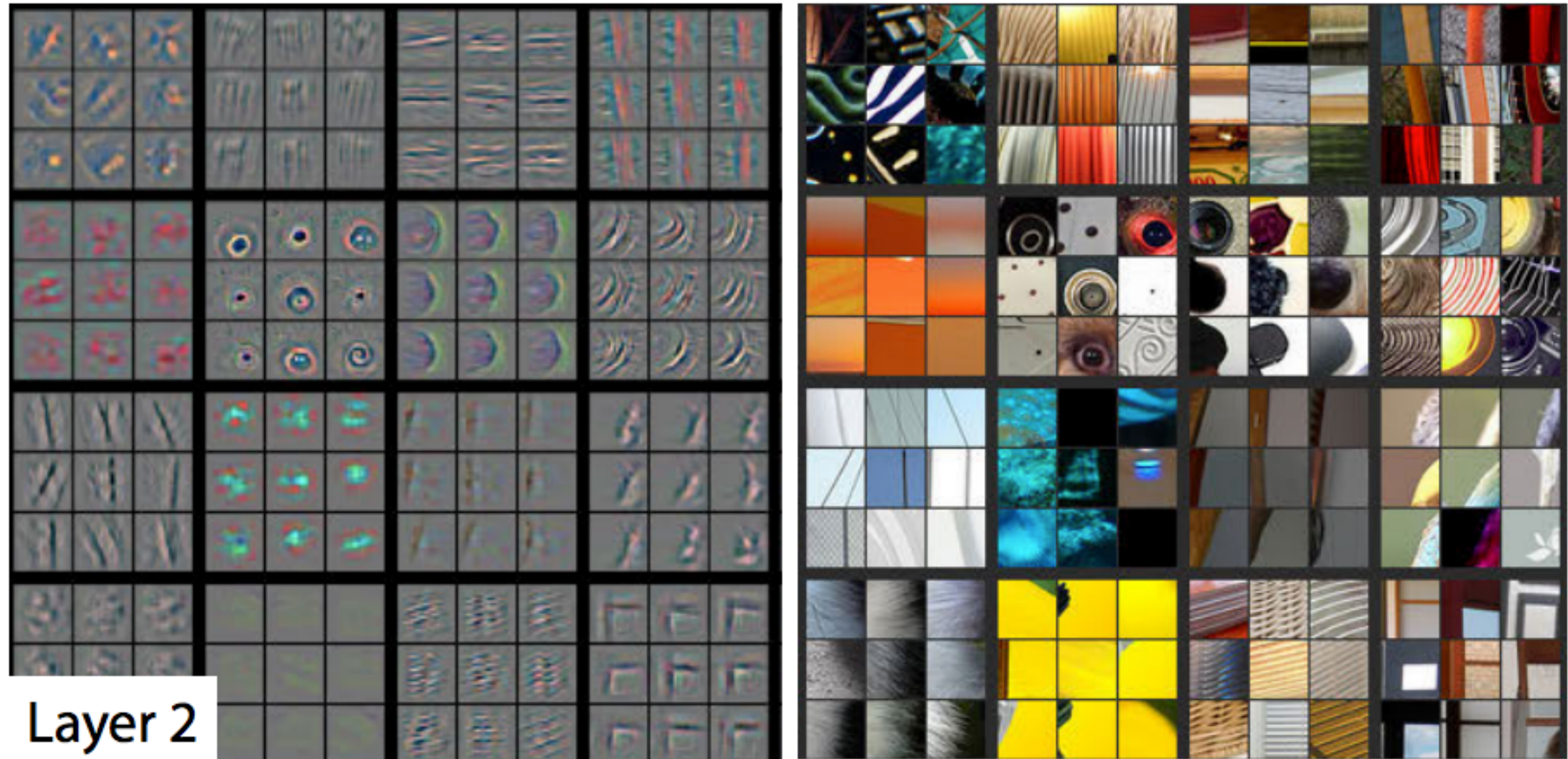
- Inspiration biologique : D. Hubel et T. Wiesel (1959, 1962, Nobel 1981)
- Cortex visuel consiste en une hiérarchie de cellules simples, complexes, et hyper-complexes



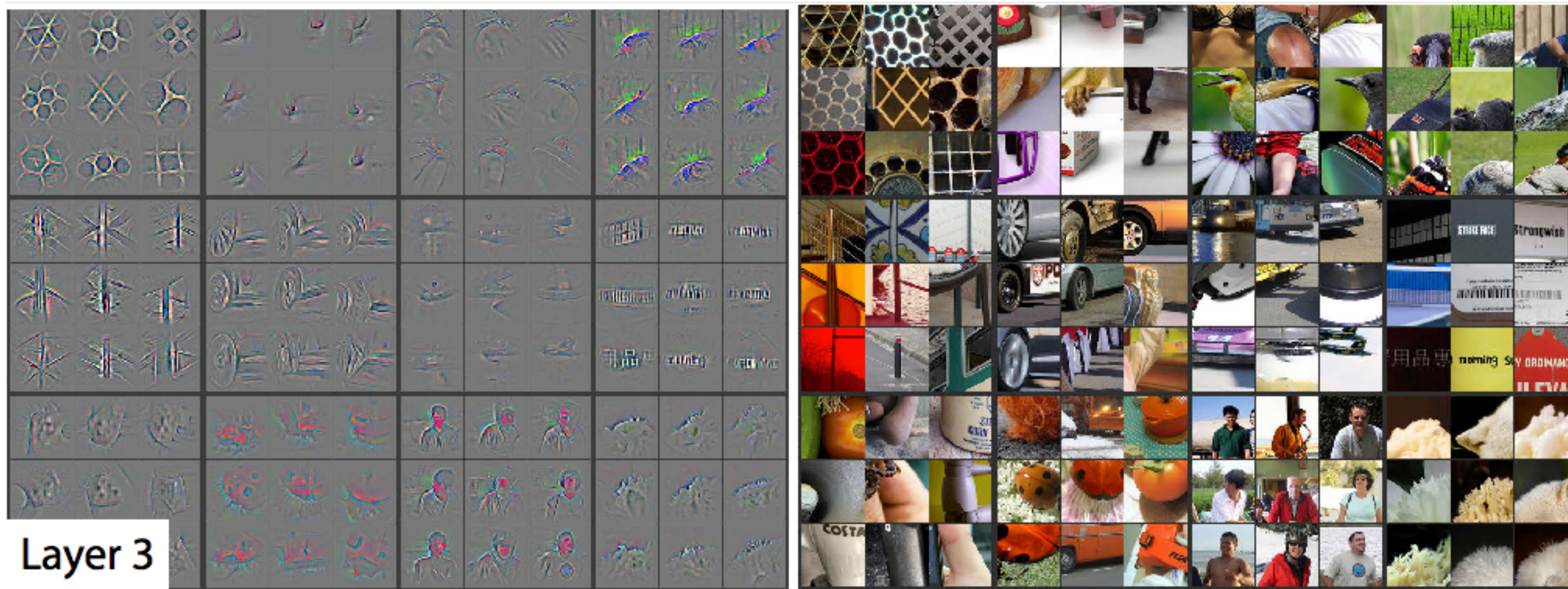
Qu'est-ce que le réseau apprend?



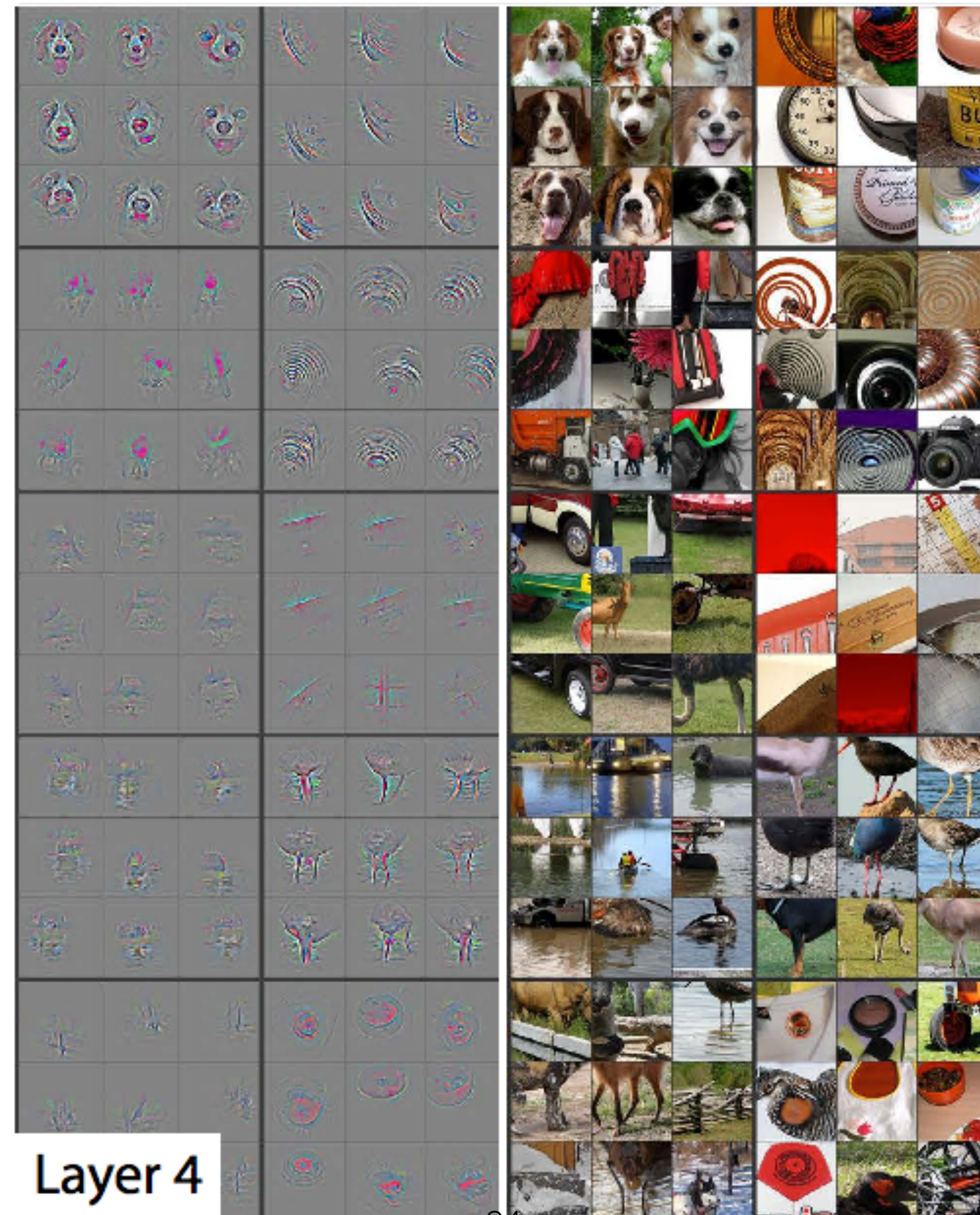
Qu'est-ce que le réseau apprend?



Qu'est-ce que le réseau apprend?

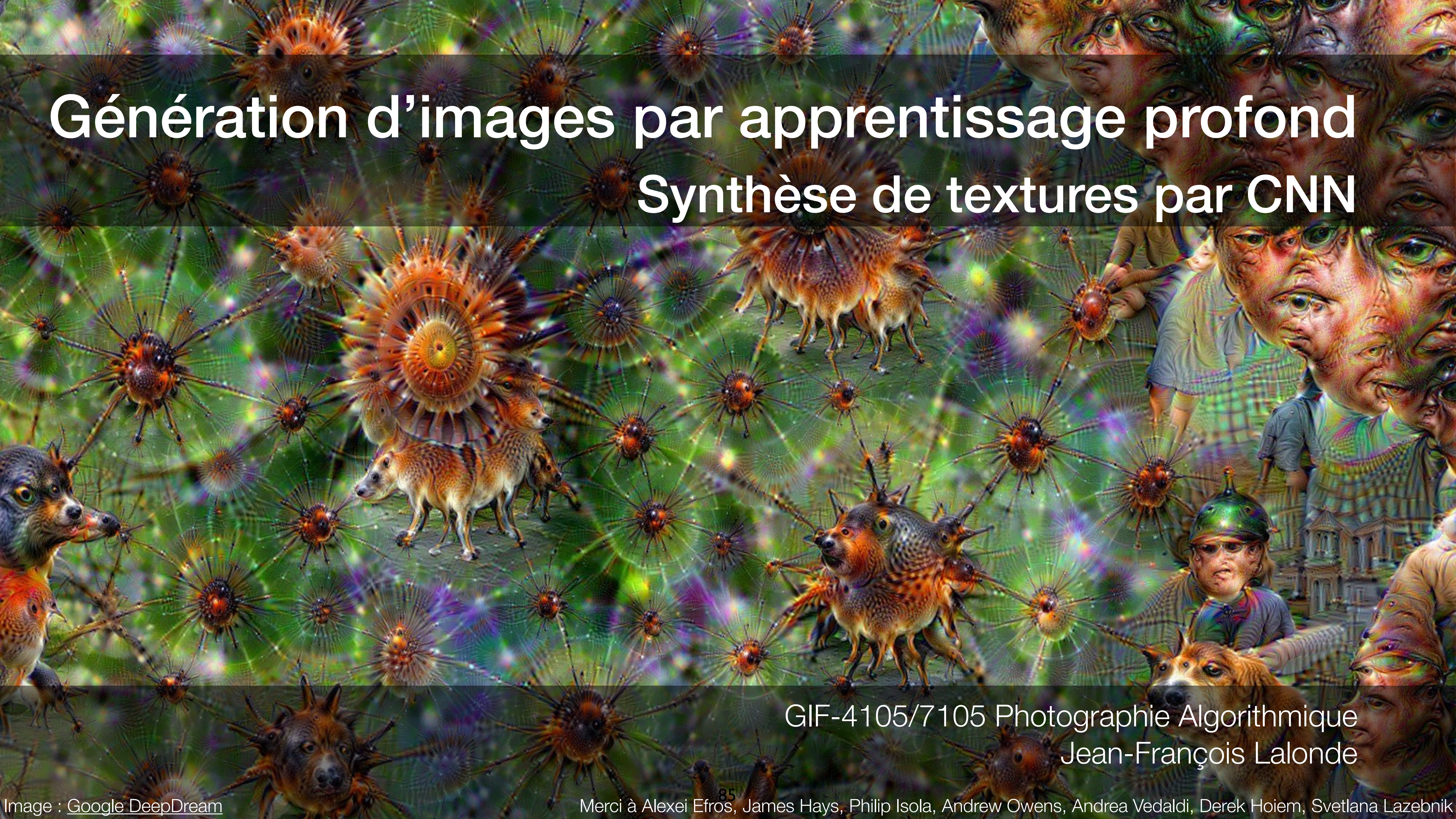


Qu'est-ce que le réseau apprend?



Génération d'images par apprentissage profond

Synthèse de textures par CNN

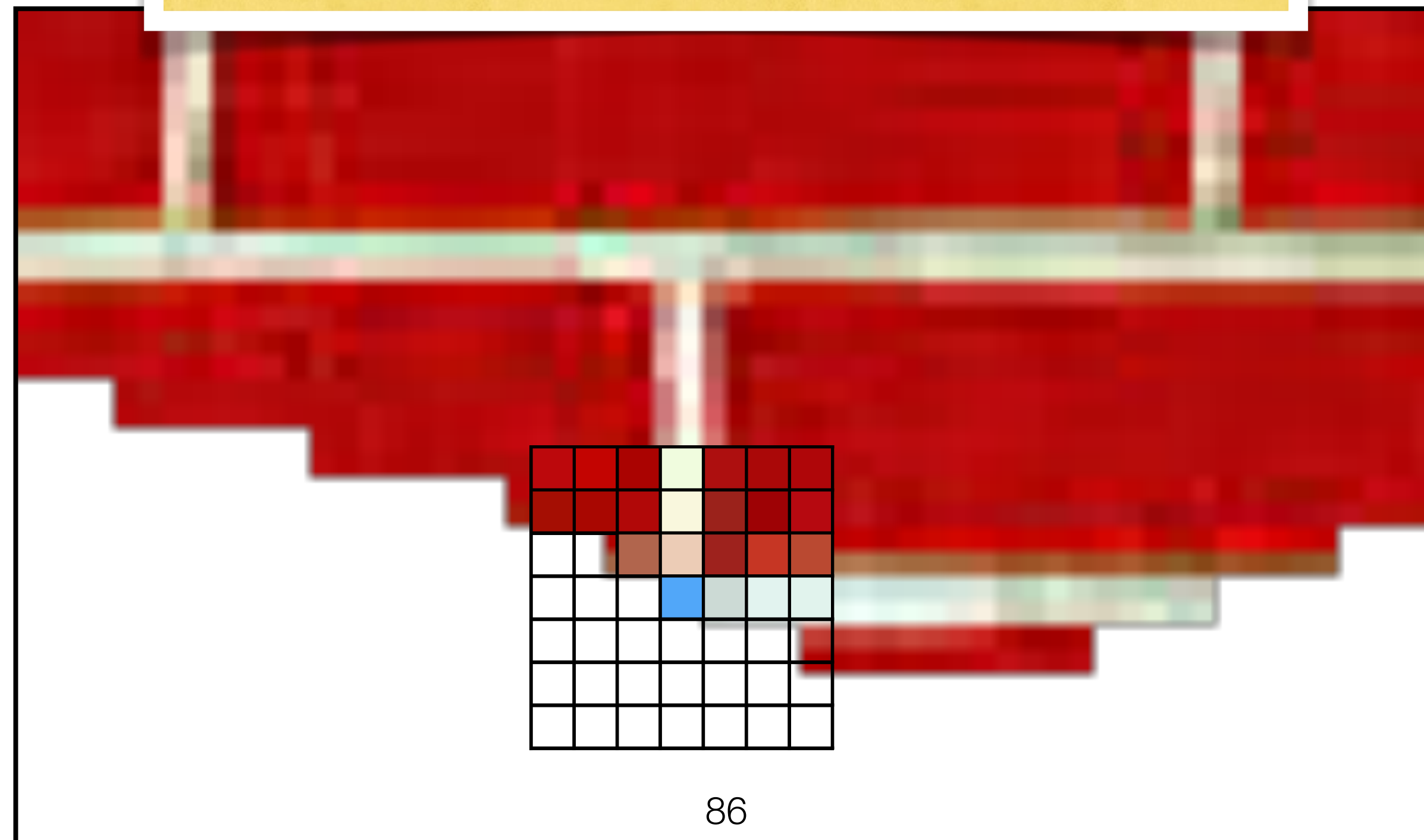


GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Idée : échantillonner l'image

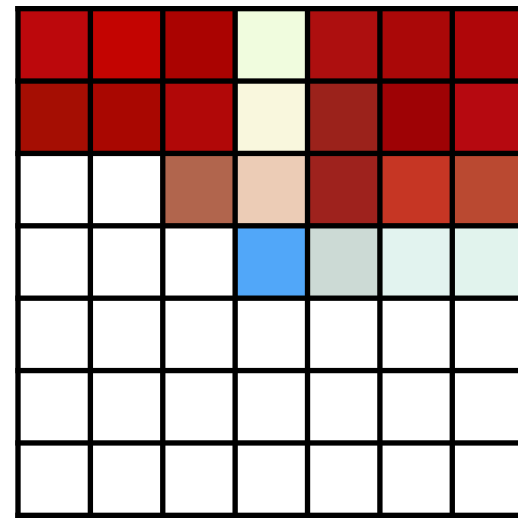
- Faisons l'hypothèse que la valeur d'un pixel ne dépend que de celles de ses voisins

Hypothèse de Markov!

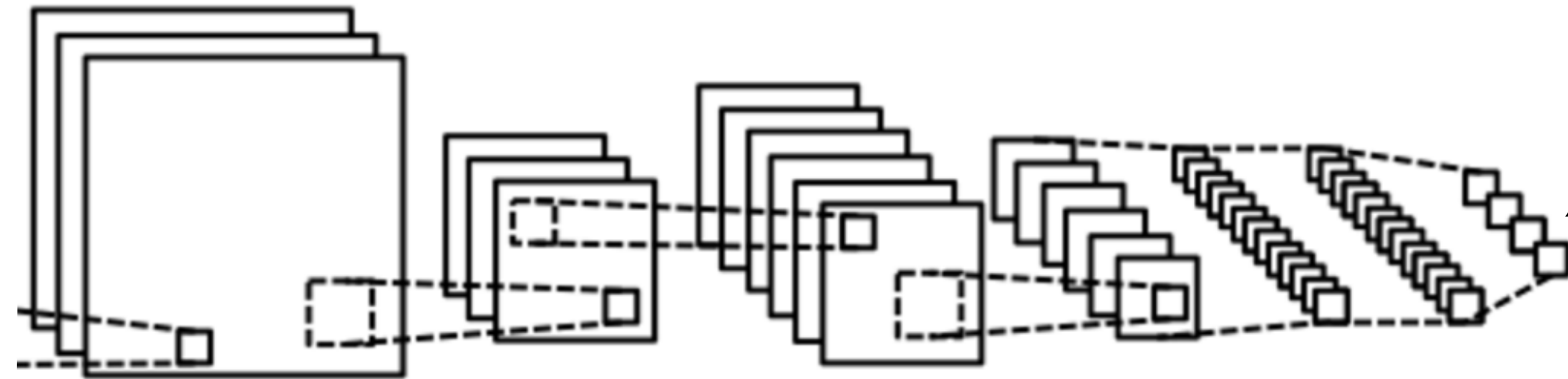


Synthèse de texture par réseau profond

Image (partielle)



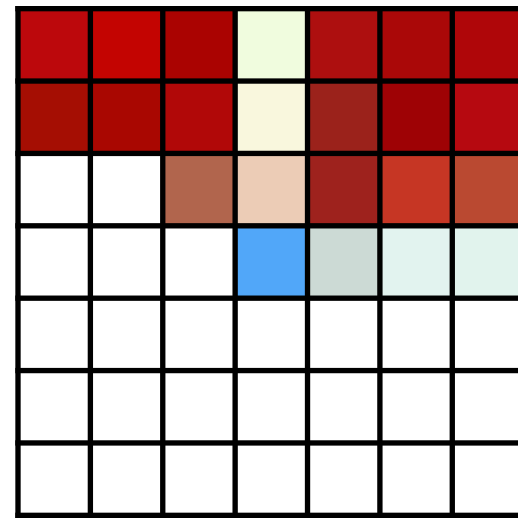
Réseau à convolutions



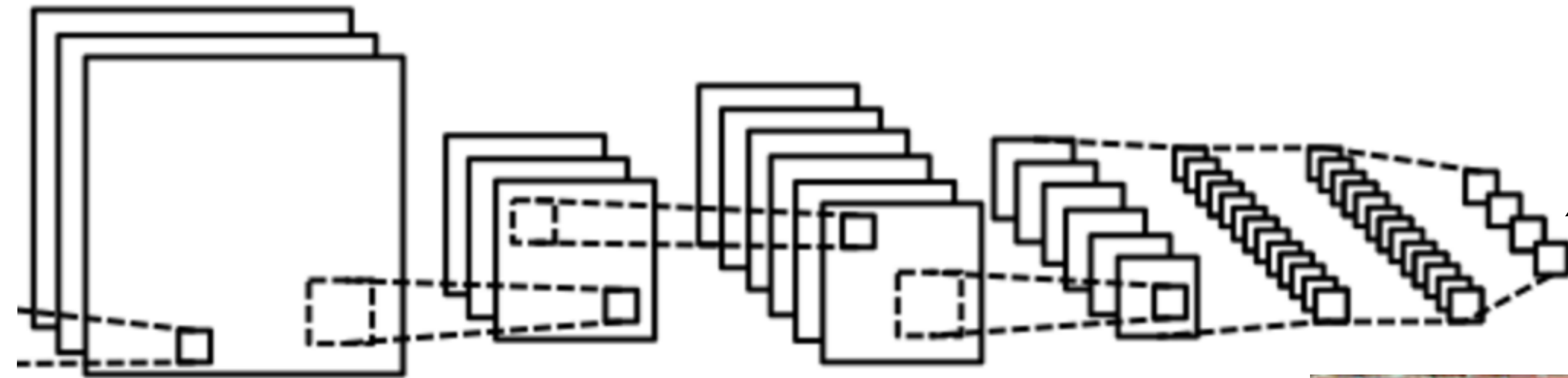
Couleur prédite du prochain pixel

Synthèse de texture par réseau profond

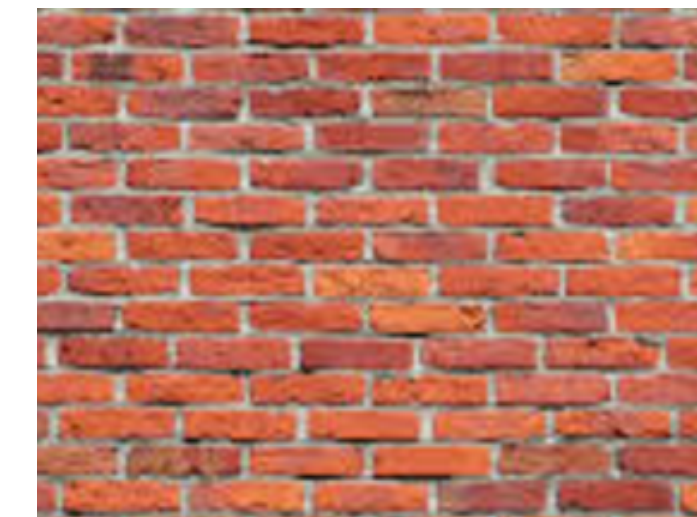
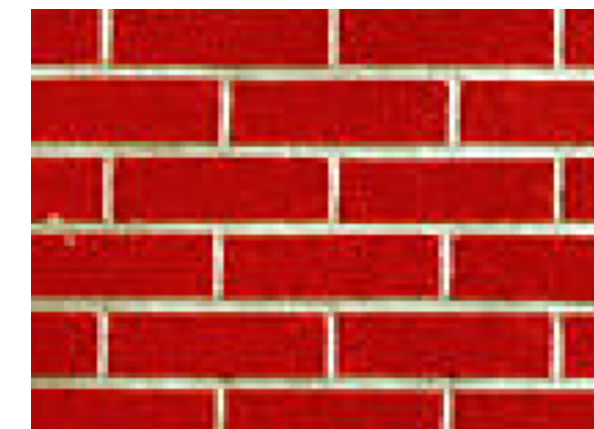
Image (partielle)



Réseau à convolutions

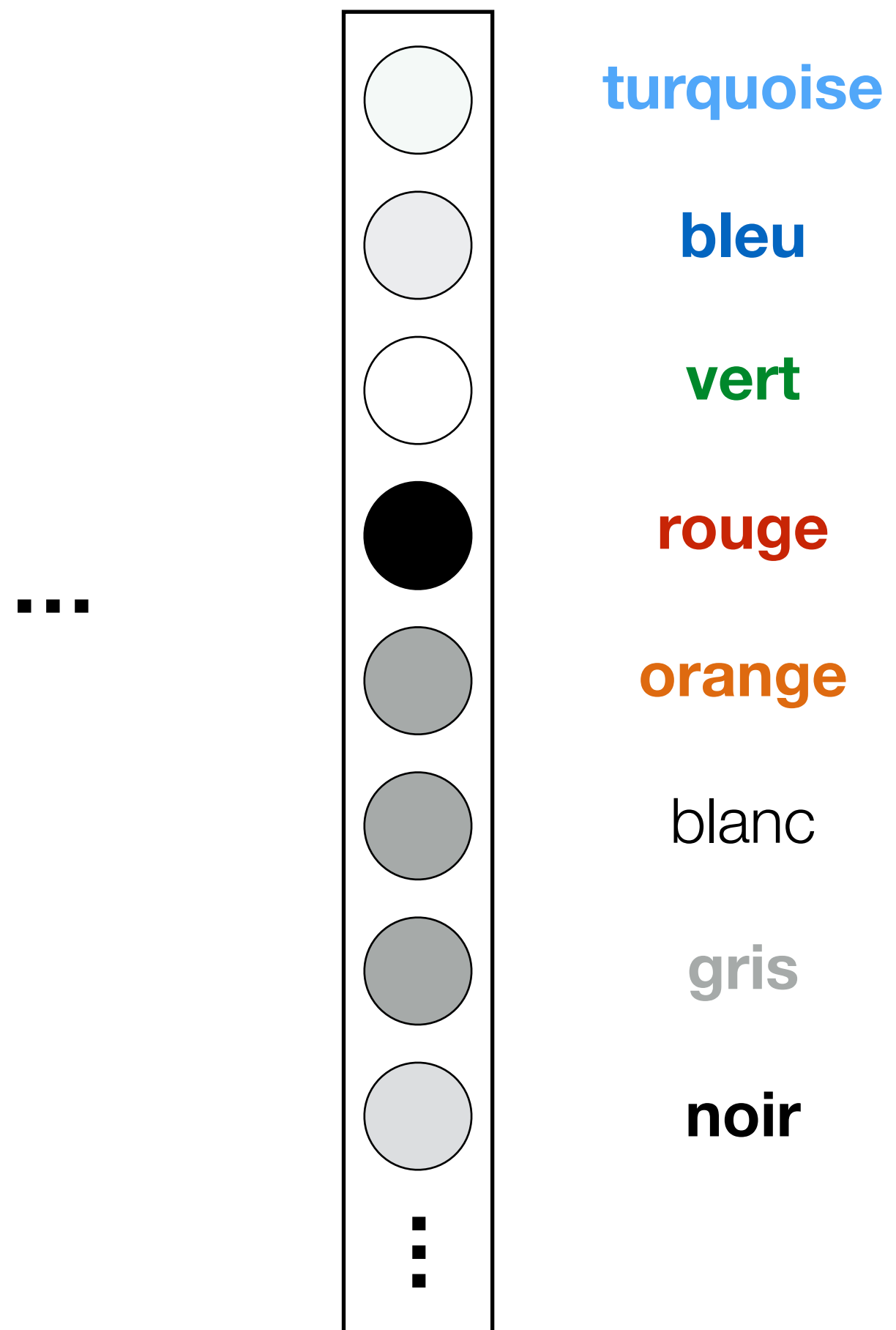


Couleur prédite du prochain pixel



Échantillonnage

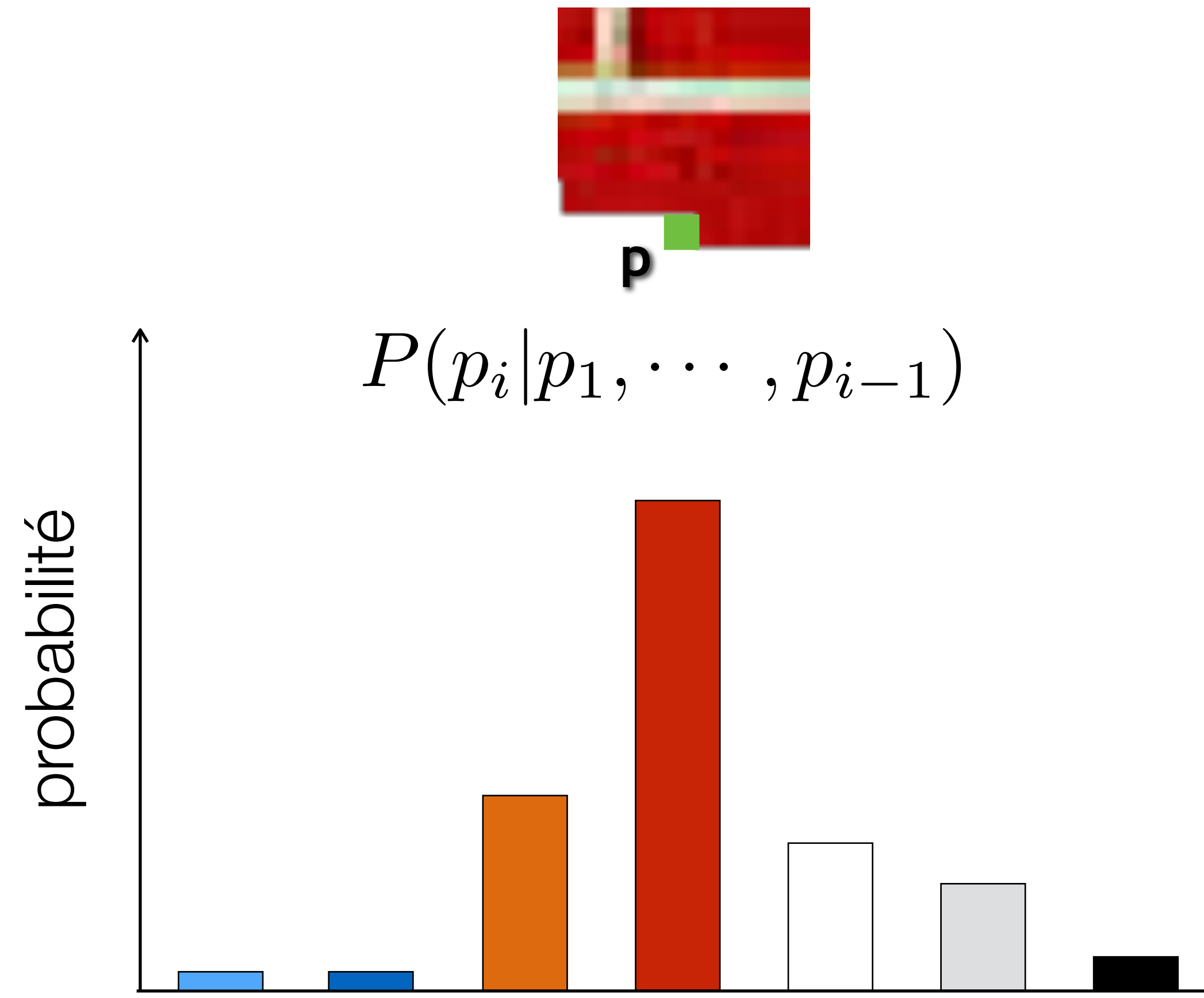
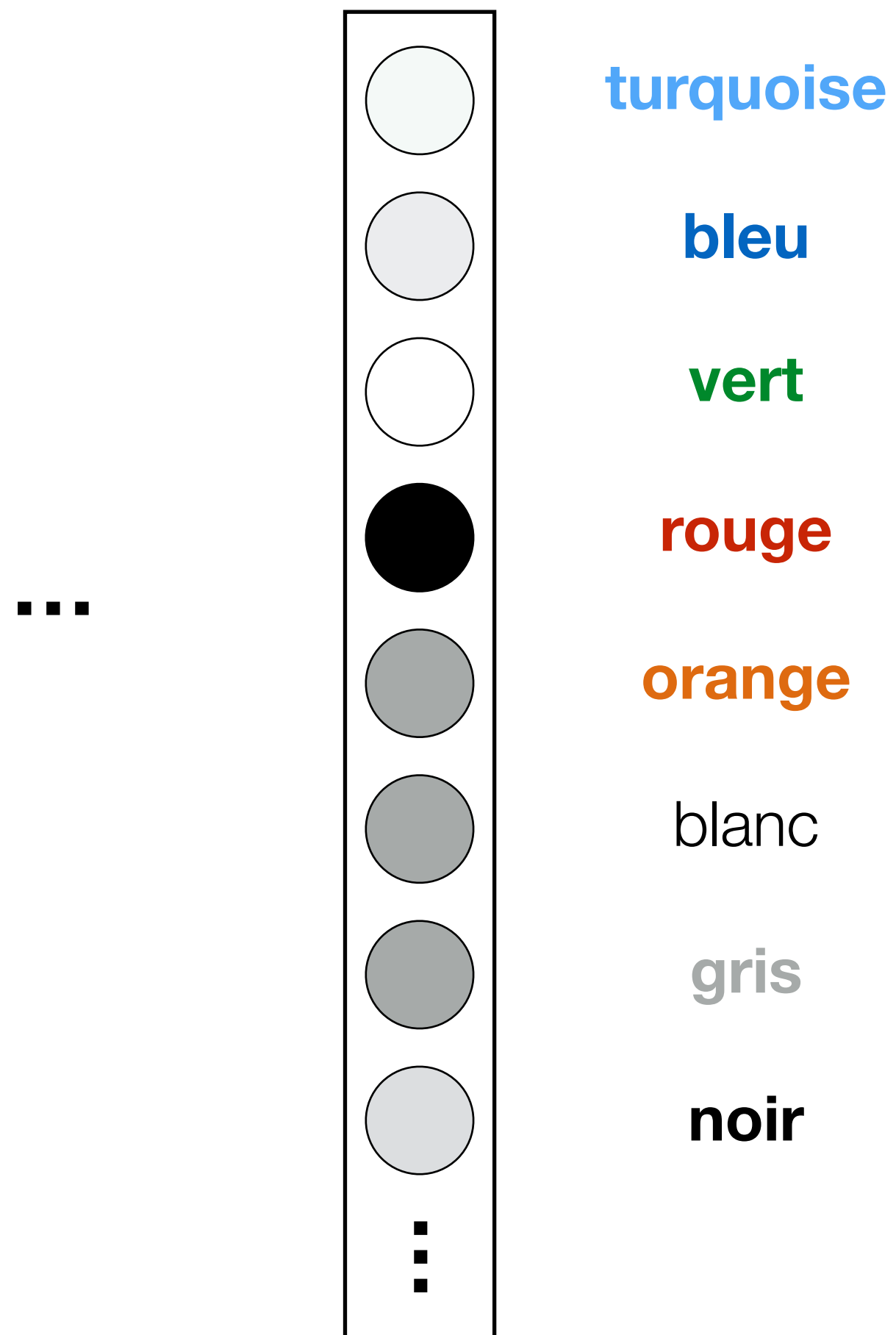
Sortie du réseau



$$P(p_i | p_1, \dots, p_{i-1})$$

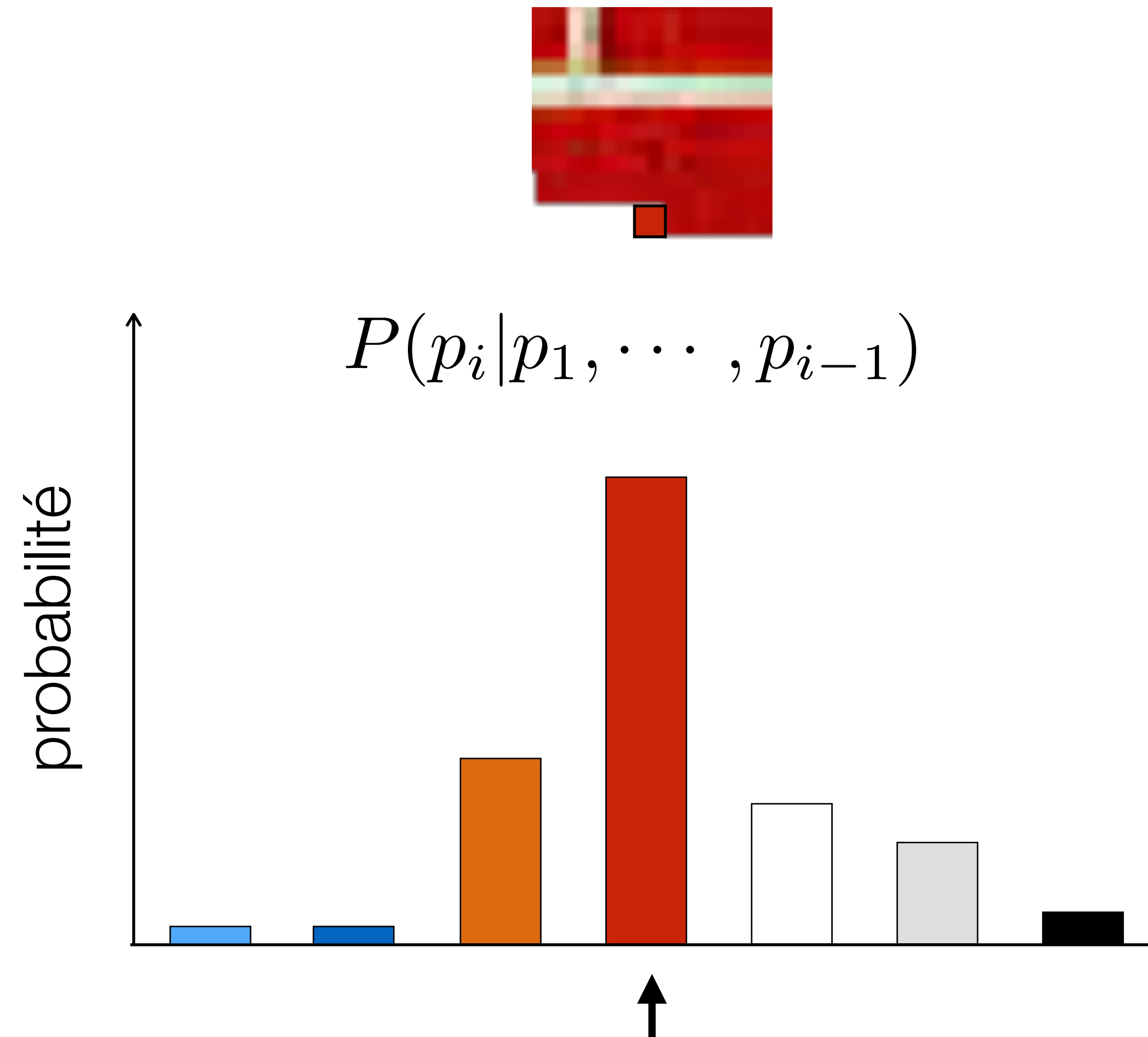
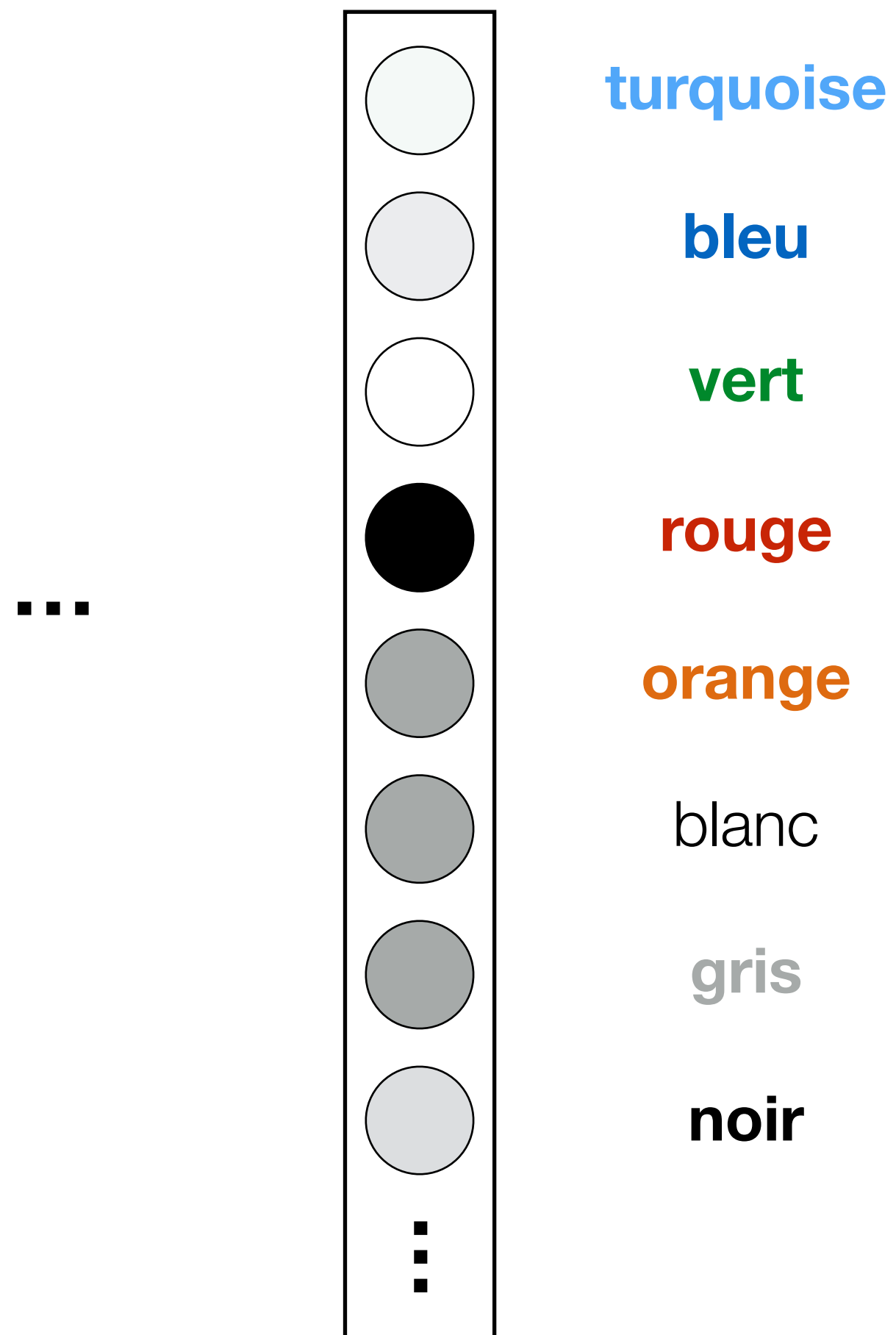
Échantillonnage

Sortie du réseau



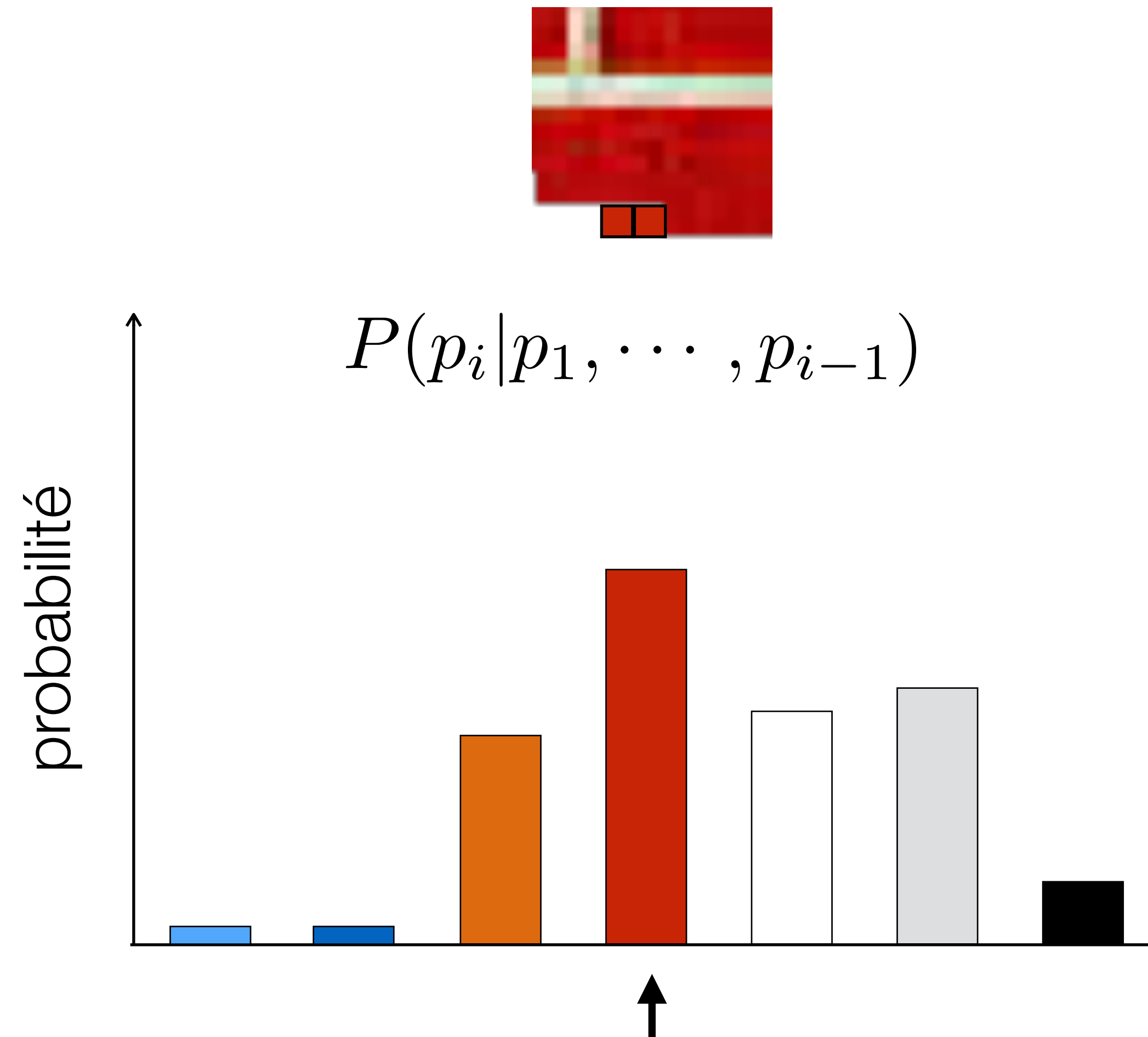
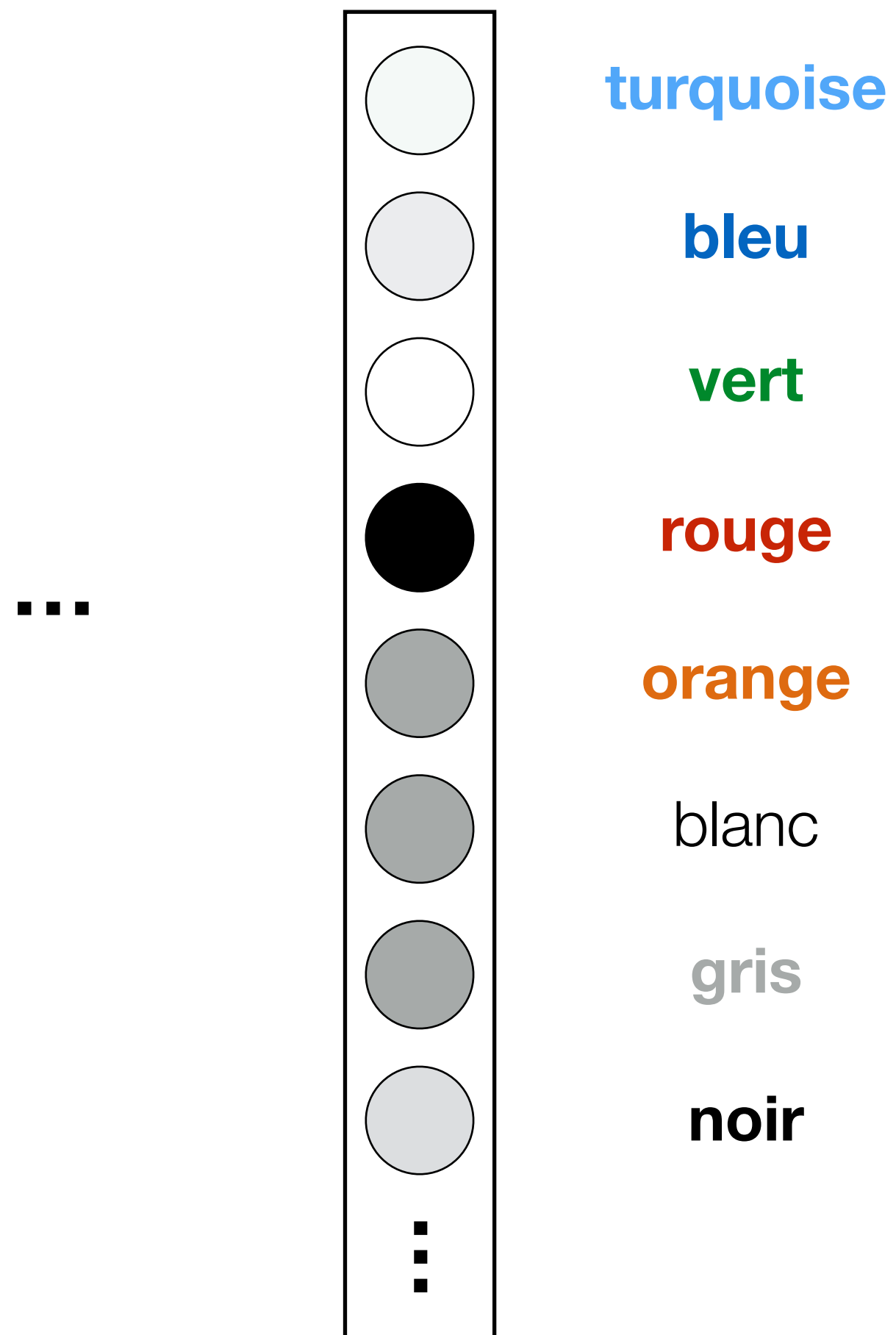
Échantillonnage

Sortie du réseau



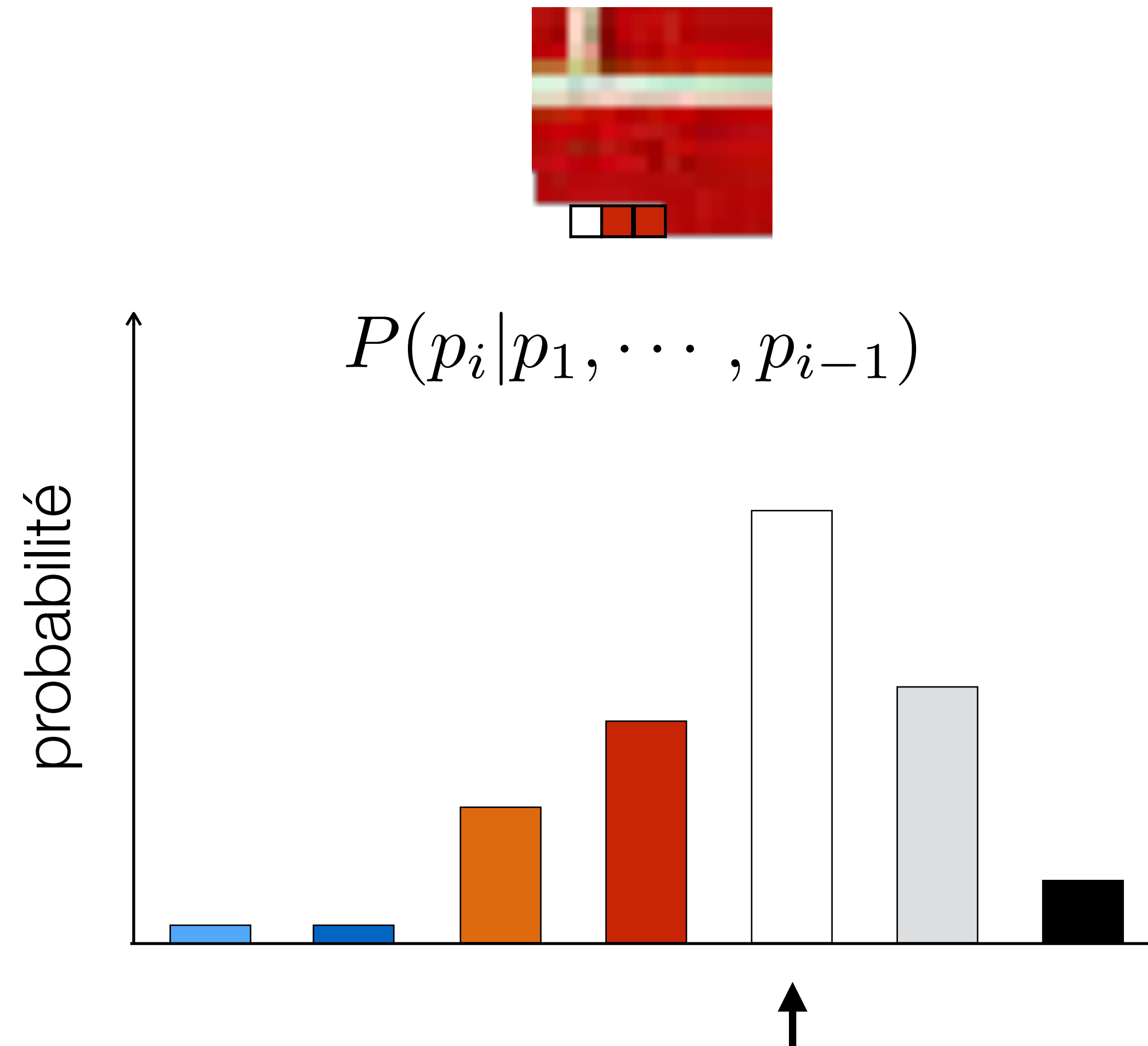
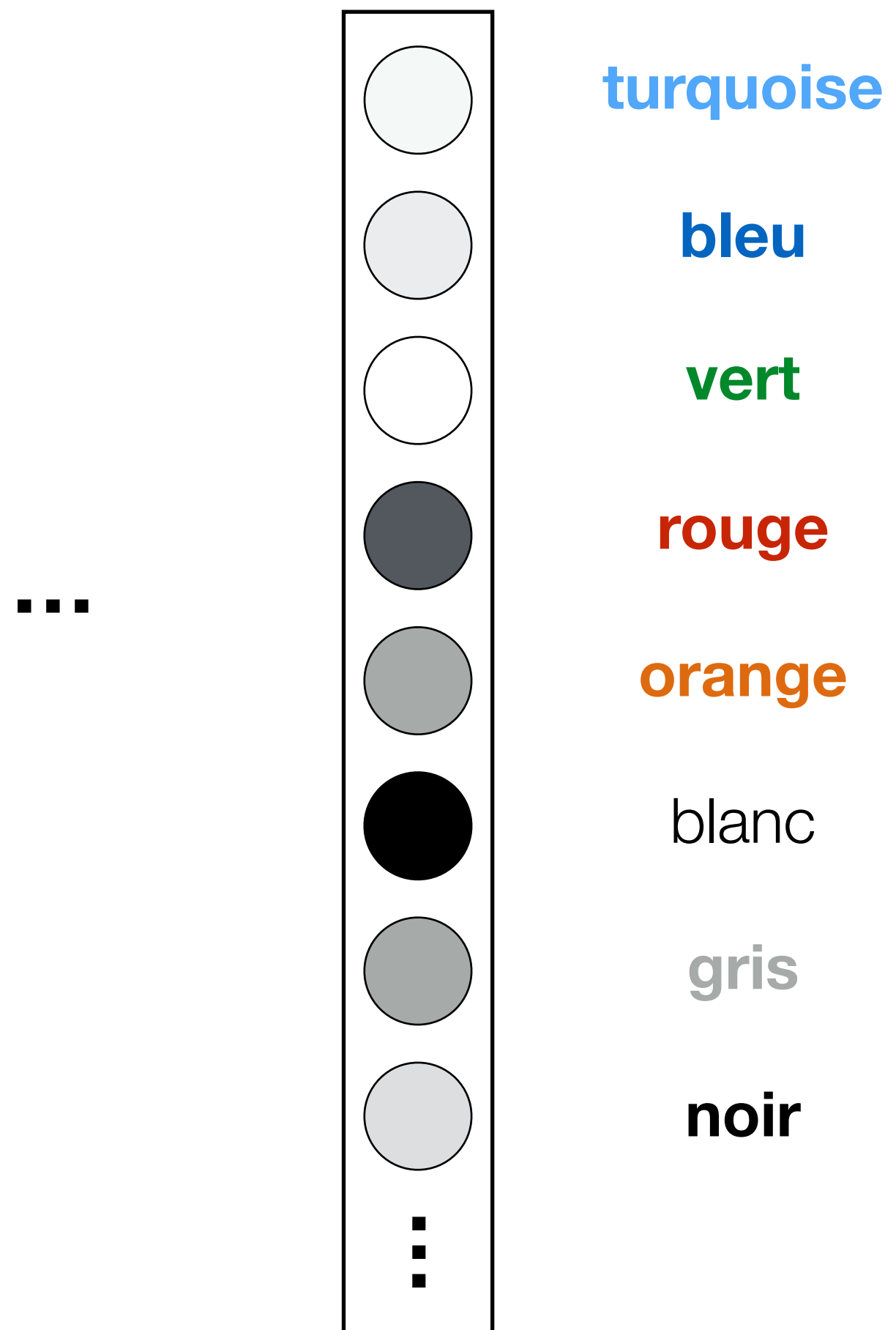
Échantillonnage

Sortie du réseau



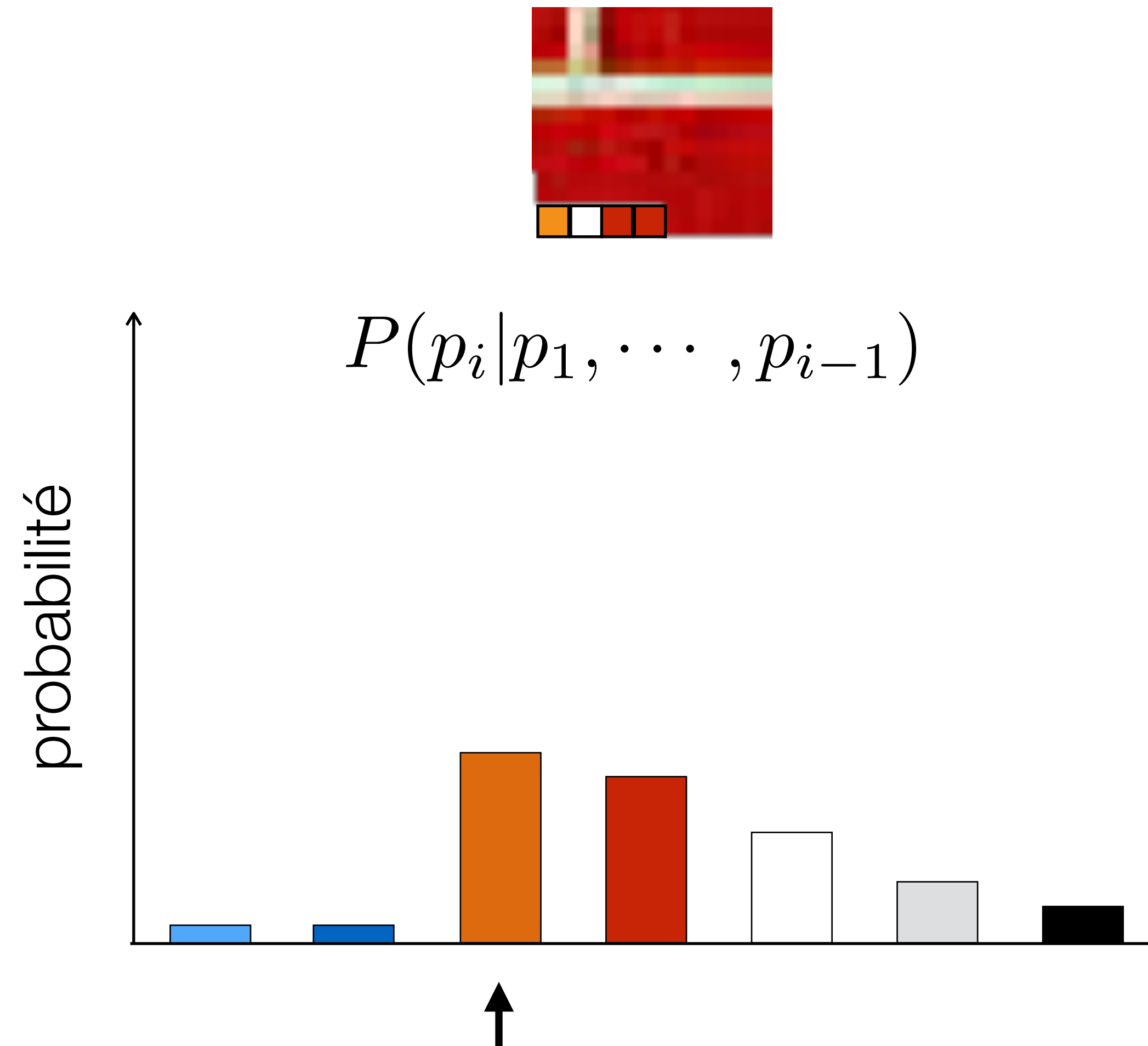
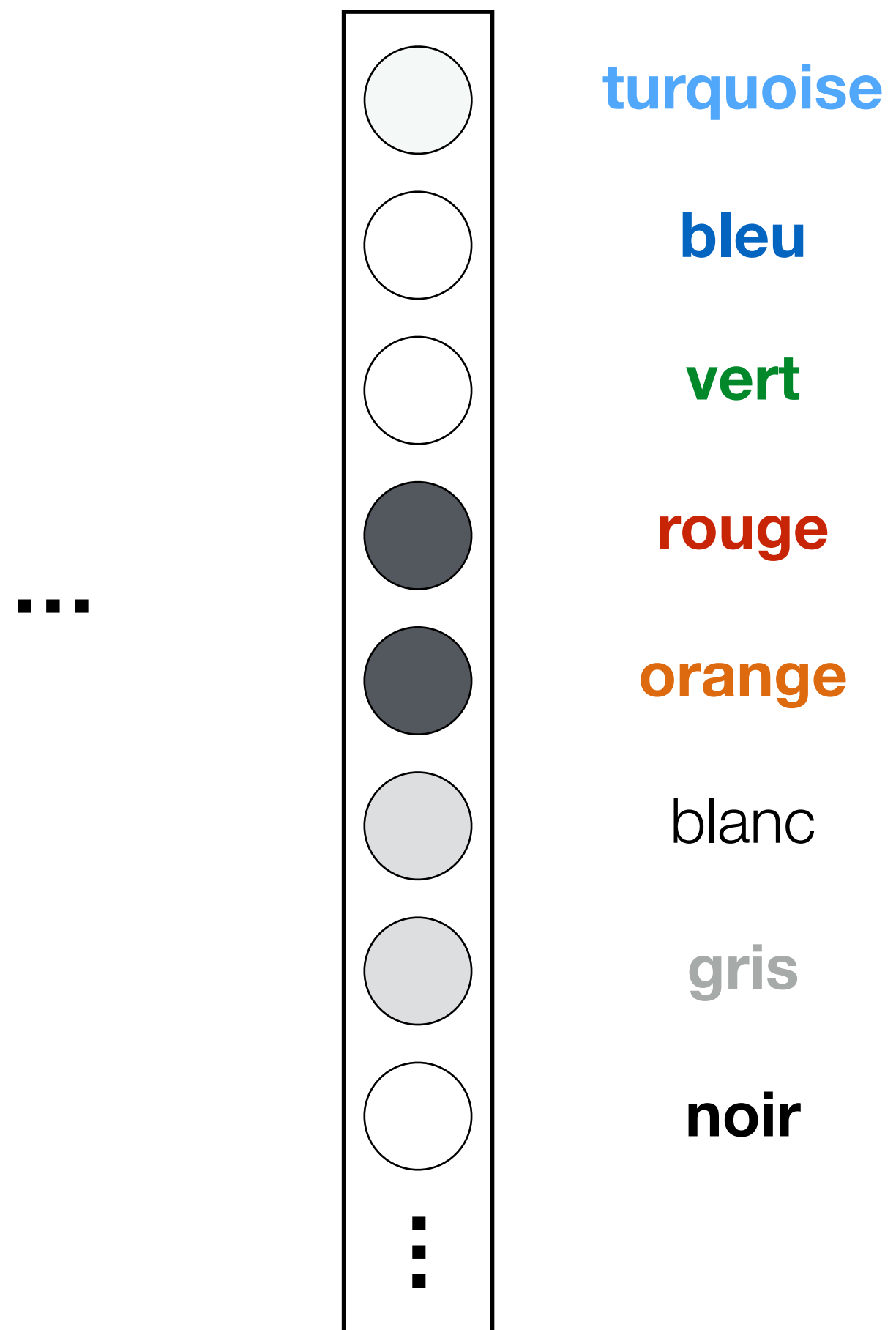
Échantillonnage

Sortie du réseau

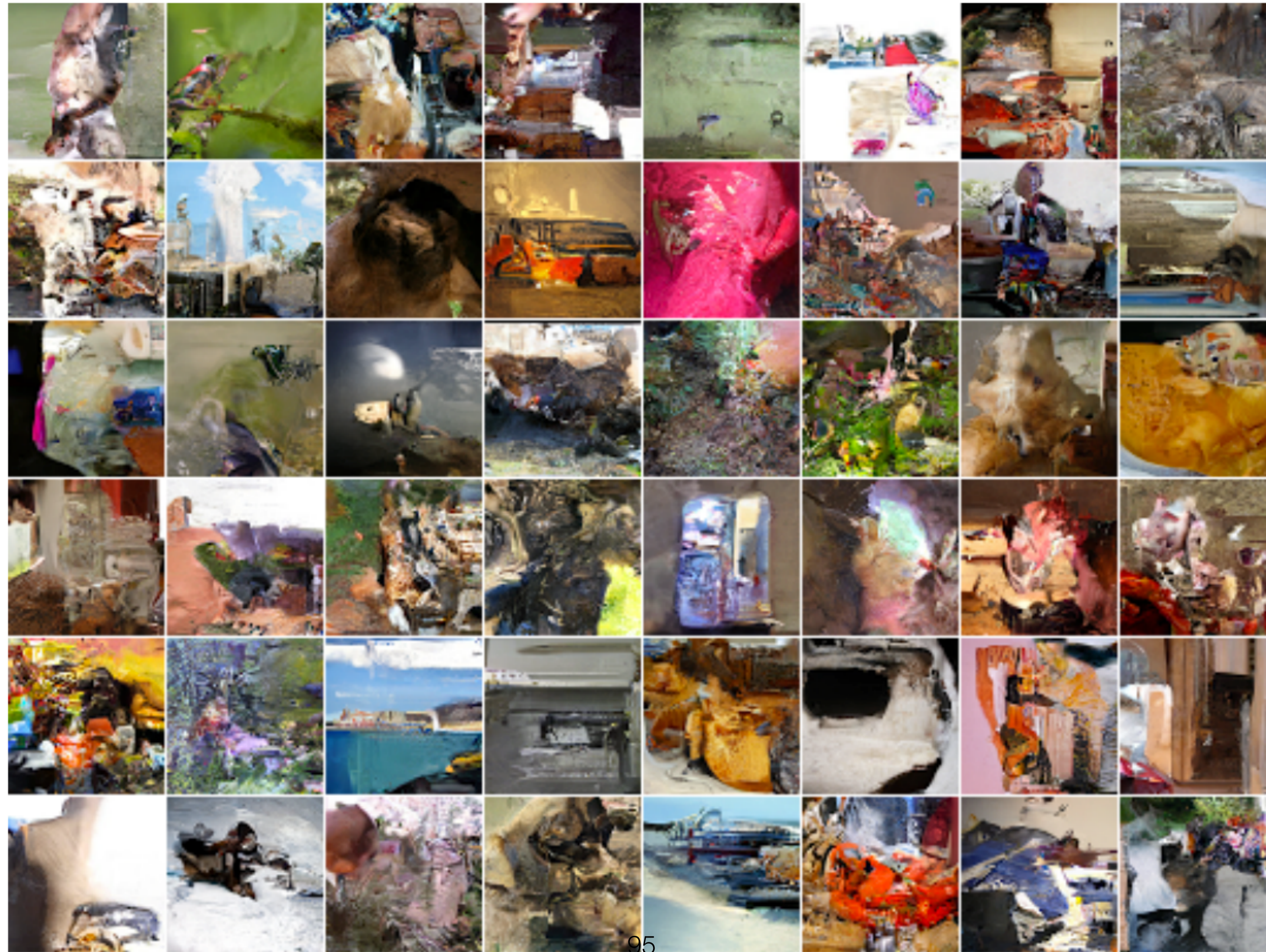


Échantillonnage

Sortie du réseau



Générer des images entières



Compléter une image

occlusion

résultats

image
originale



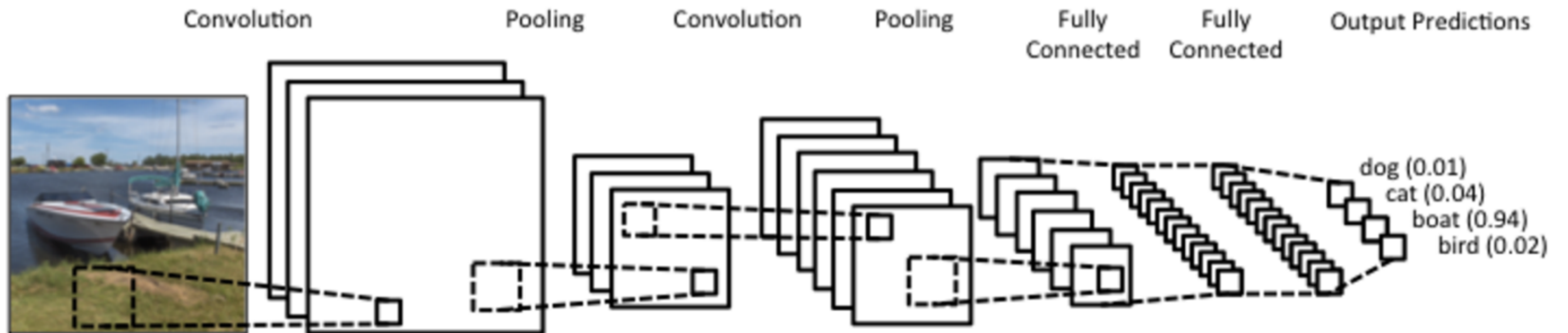
Génération d'images par apprentissage profond

Synthèse de textures par appariement de caractéristiques profondes

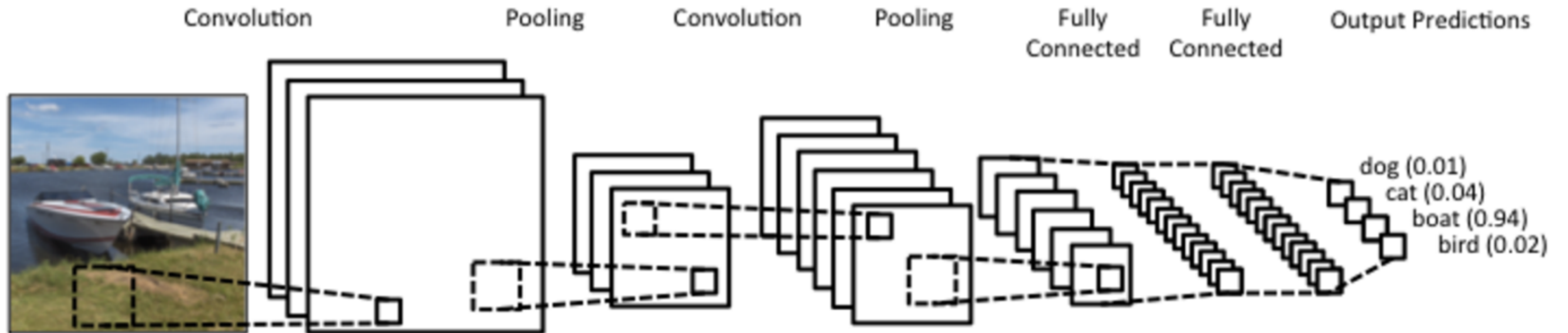


GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

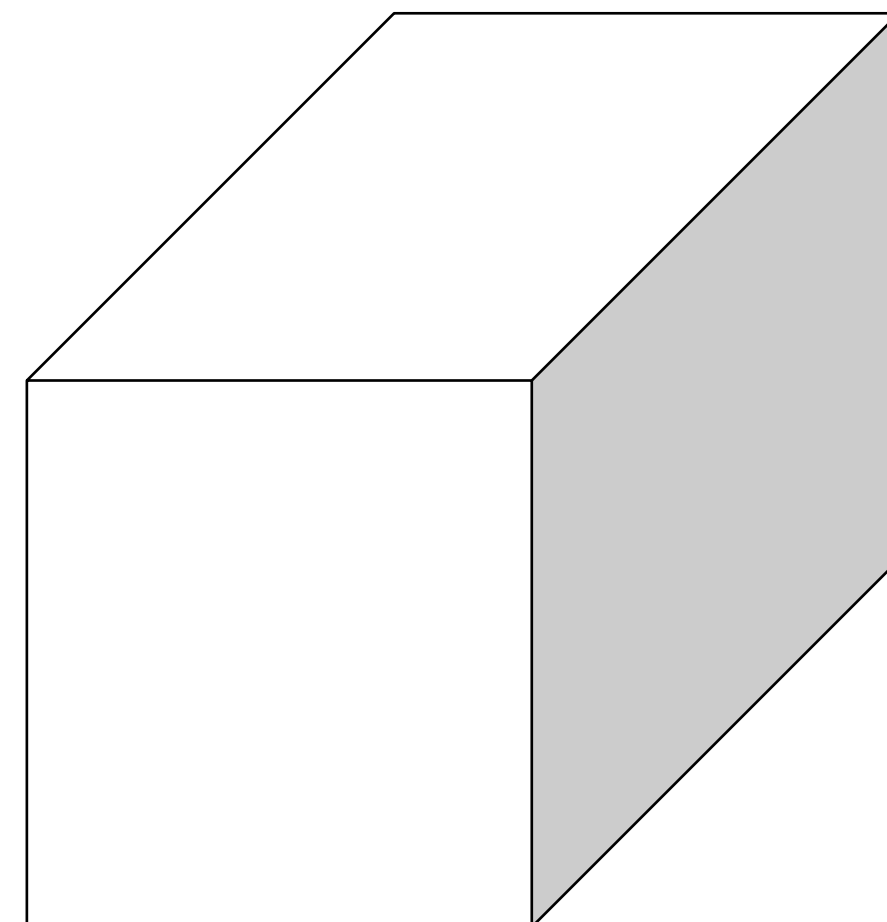
L'architecture d'un réseau à convolutions



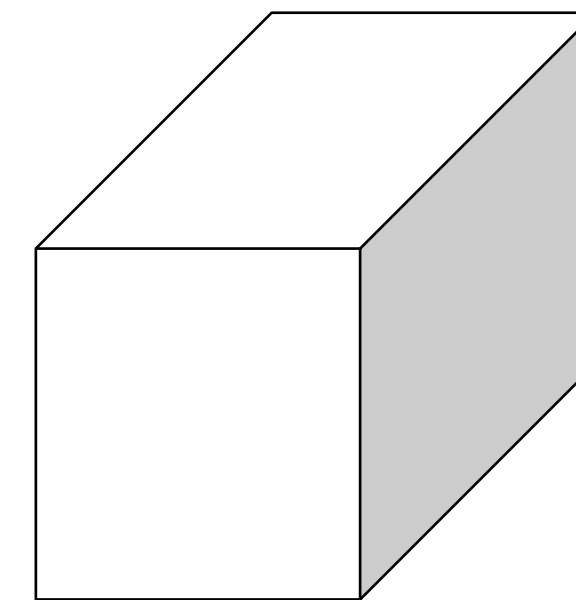
L'architecture d'un réseau à convolutions



Carte de caractéristiques
échelle 1



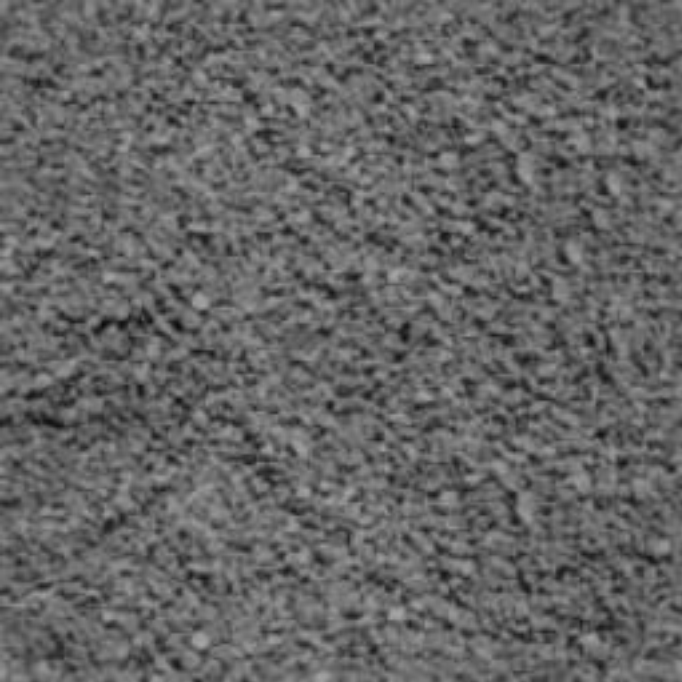
Carte de caractéristiques
échelle 2



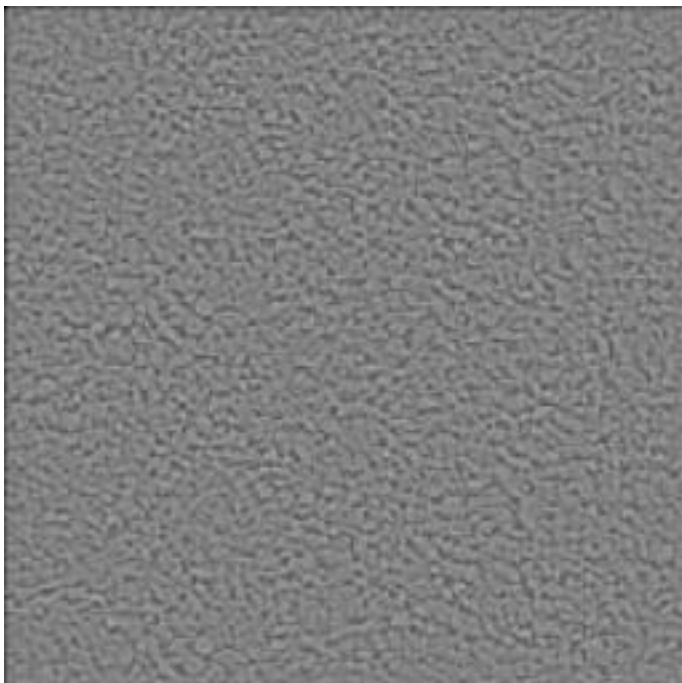
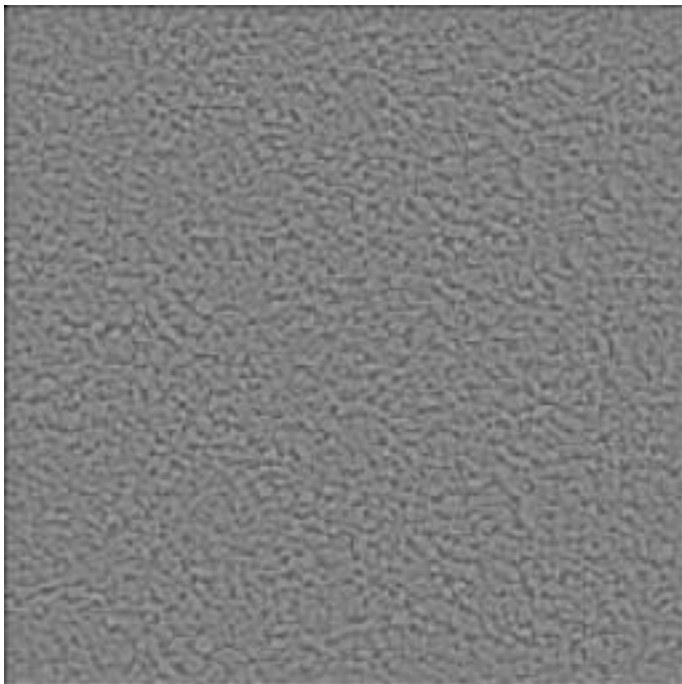
Observation

Un réseau de neurone entraîné à reconnaître les objets devient un calculateur de caractéristiques multi-échelle très robustes.

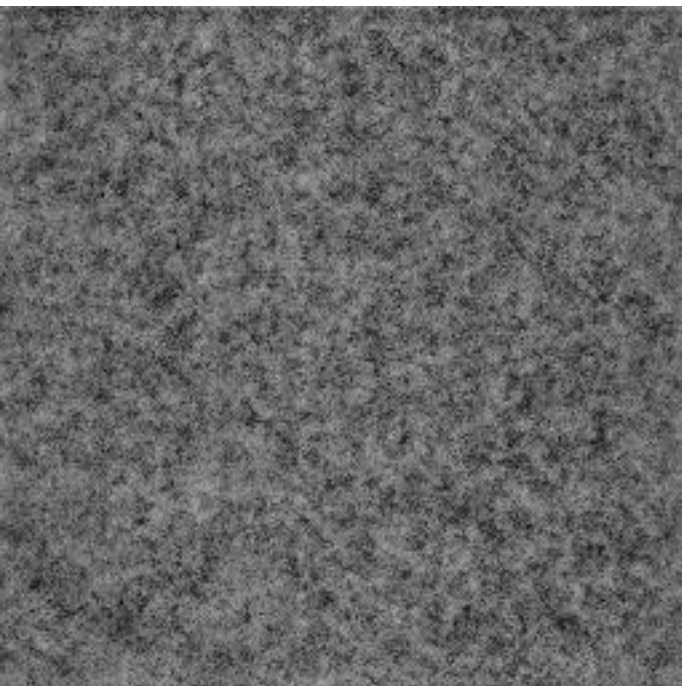
Exemple de texture



Niveau 0



Texture générée

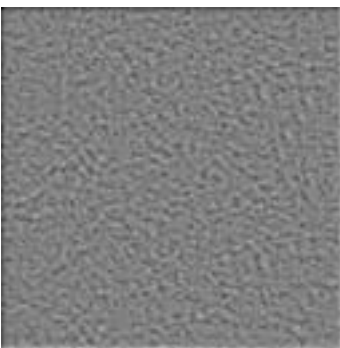
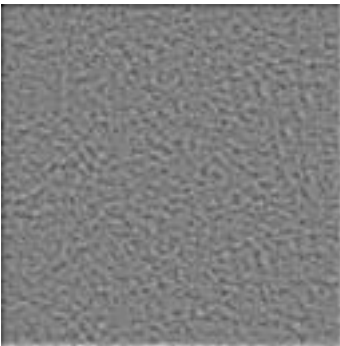


Statistiques

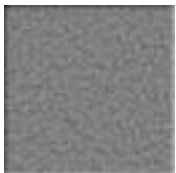
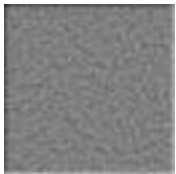
Histogrammes de différentes bandes de fréquences

Rappel

Niveau 1



Niveau 2



Niveau 3

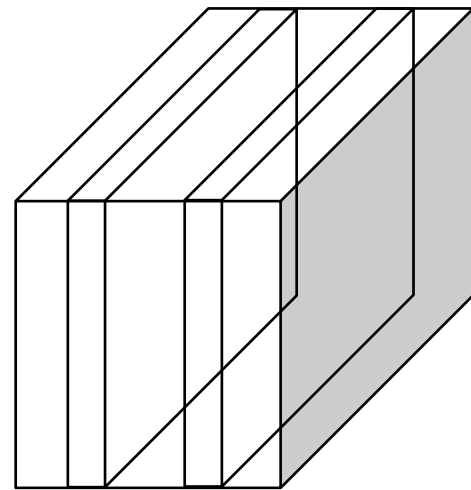


Après l'itération 1

Synthèse de textures par CNN

$$G_{ij} = \sum_{x=1}^w \sum_{y=1}^h c_i(x, y) c_j(x, y)$$

Caractéristiques



Couche de convolutions

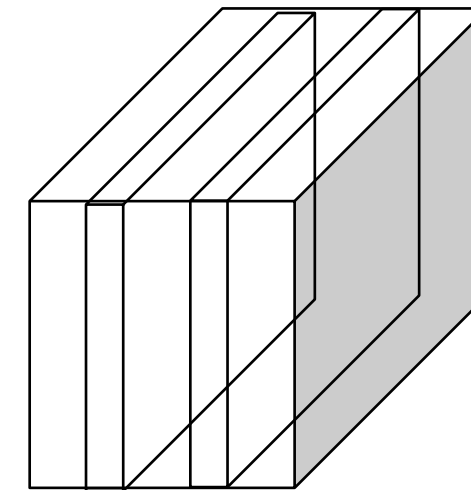


Image source

Calcule la corrélation entre les caractéristiques

$$\hat{G}_{ij} = \sum_{x=1}^w \sum_{y=1}^h \hat{c}_i(x, y) \hat{c}_j(x, y)$$

Caractéristiques



Couche de convolutions

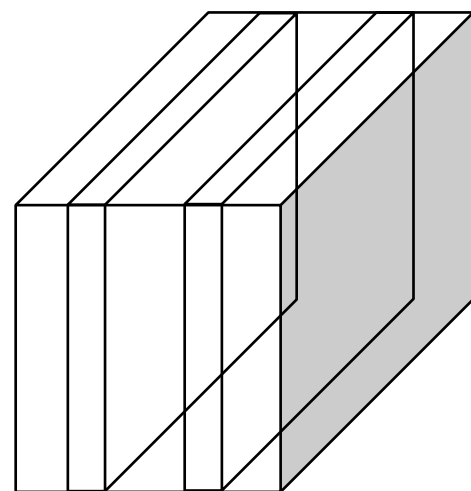


Image générée

Synthèse de textures par CNN

$$G_{ij} = \sum_{x=1}^w \sum_{y=1}^h c_i(x, y) c_j(x, y)$$

Caractéristiques



Couche de convolutions



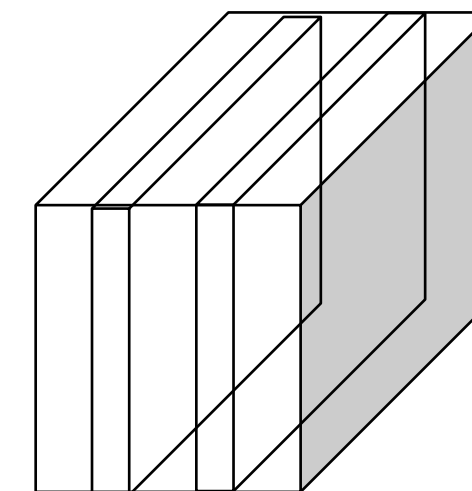
Image source

Minimise la différence entre les corrélations pour reproduire la texture!

$$\sum_{i,j} (G_{ij} - \hat{G}_{ij})^2$$

$$\hat{G}_{ij} = \sum_{x=1}^w \sum_{y=1}^h \hat{c}_i(x, y) \hat{c}_j(x, y)$$

Caractéristiques



Couche de convolutions

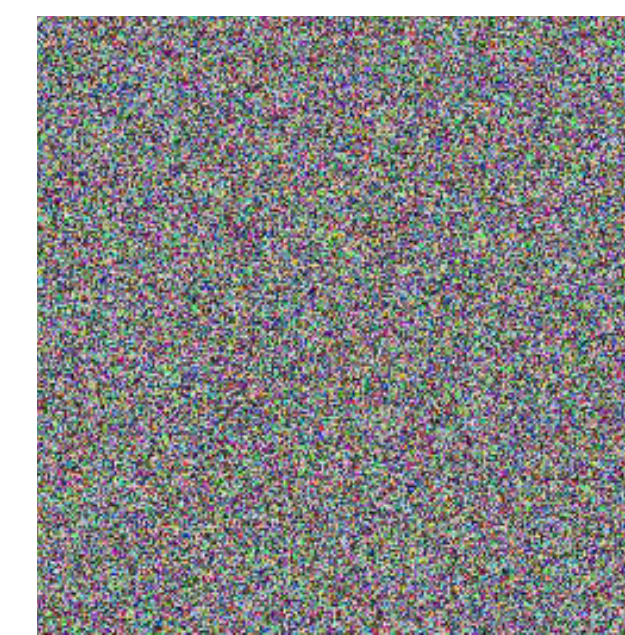
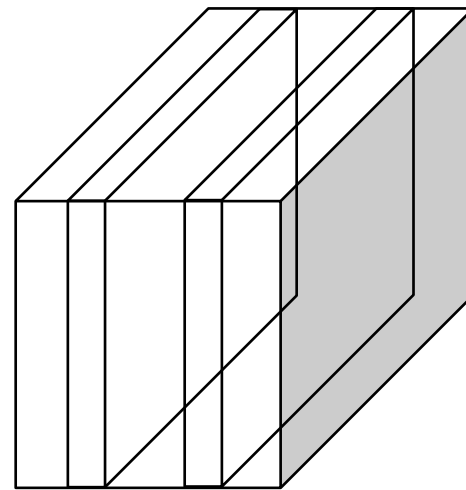


Image générée

Synthèse de textures par CNN

$$G_{ij} = \sum_{x=1}^w \sum_{y=1}^h c_i(x, y) c_j(x, y)$$

Caractéristiques



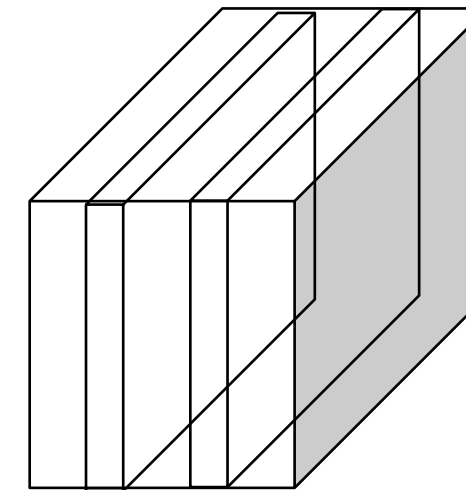
Couche de convolutions



Image source

$$\hat{G}_{ij} = \sum_{x=1}^w \sum_{y=1}^h \hat{c}_i(x, y) \hat{c}_j(x, y)$$

Caractéristiques



Couche de convolutions

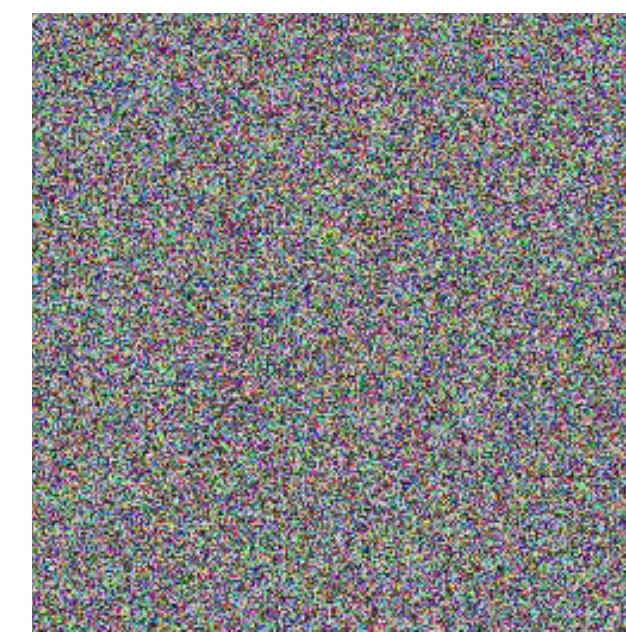
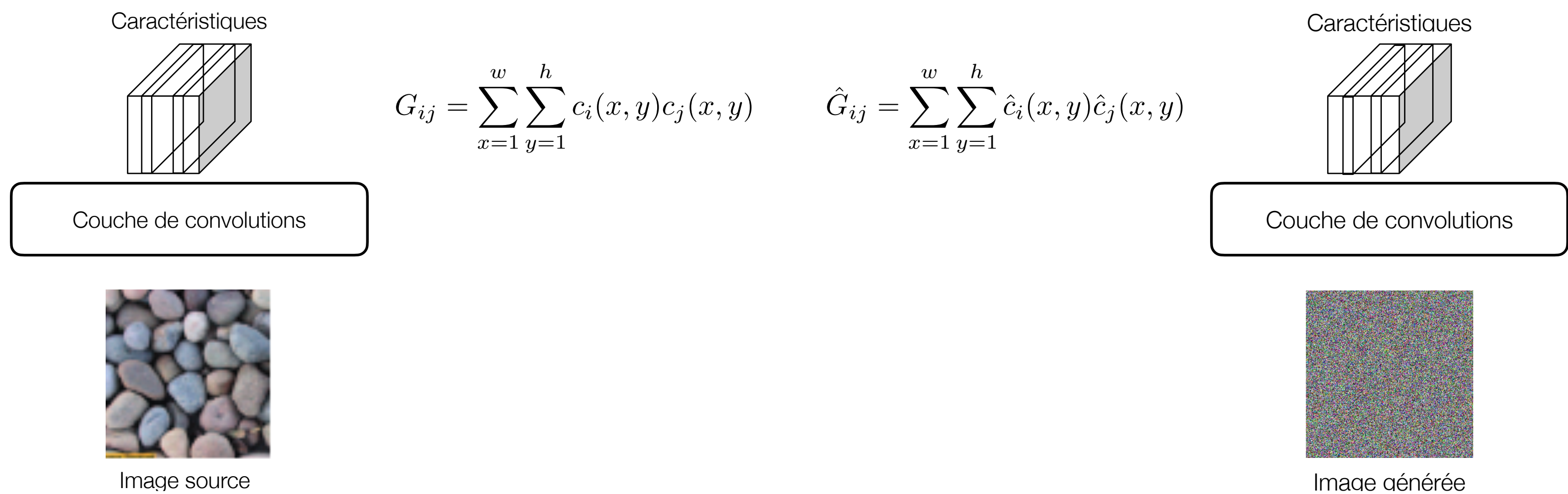
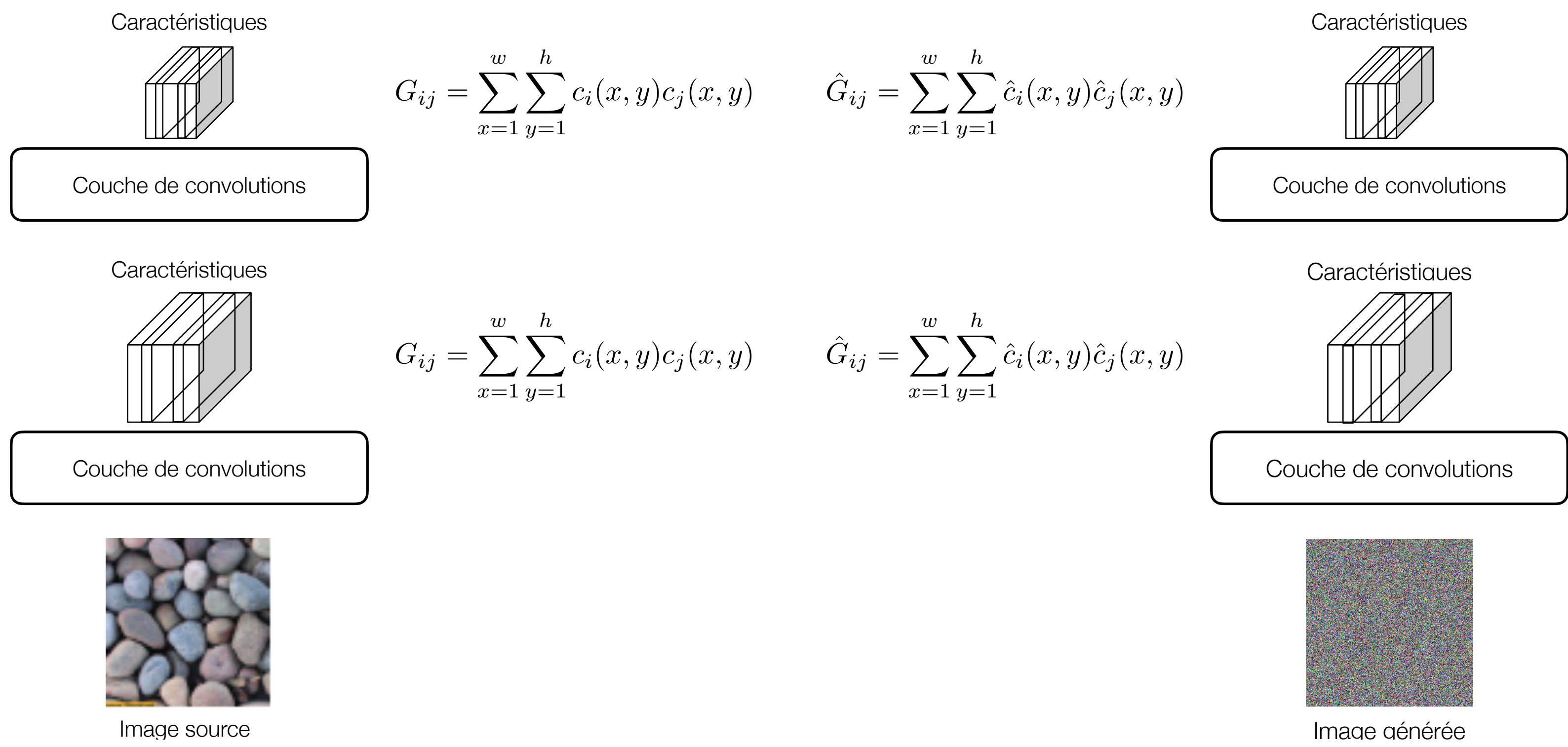


Image générée

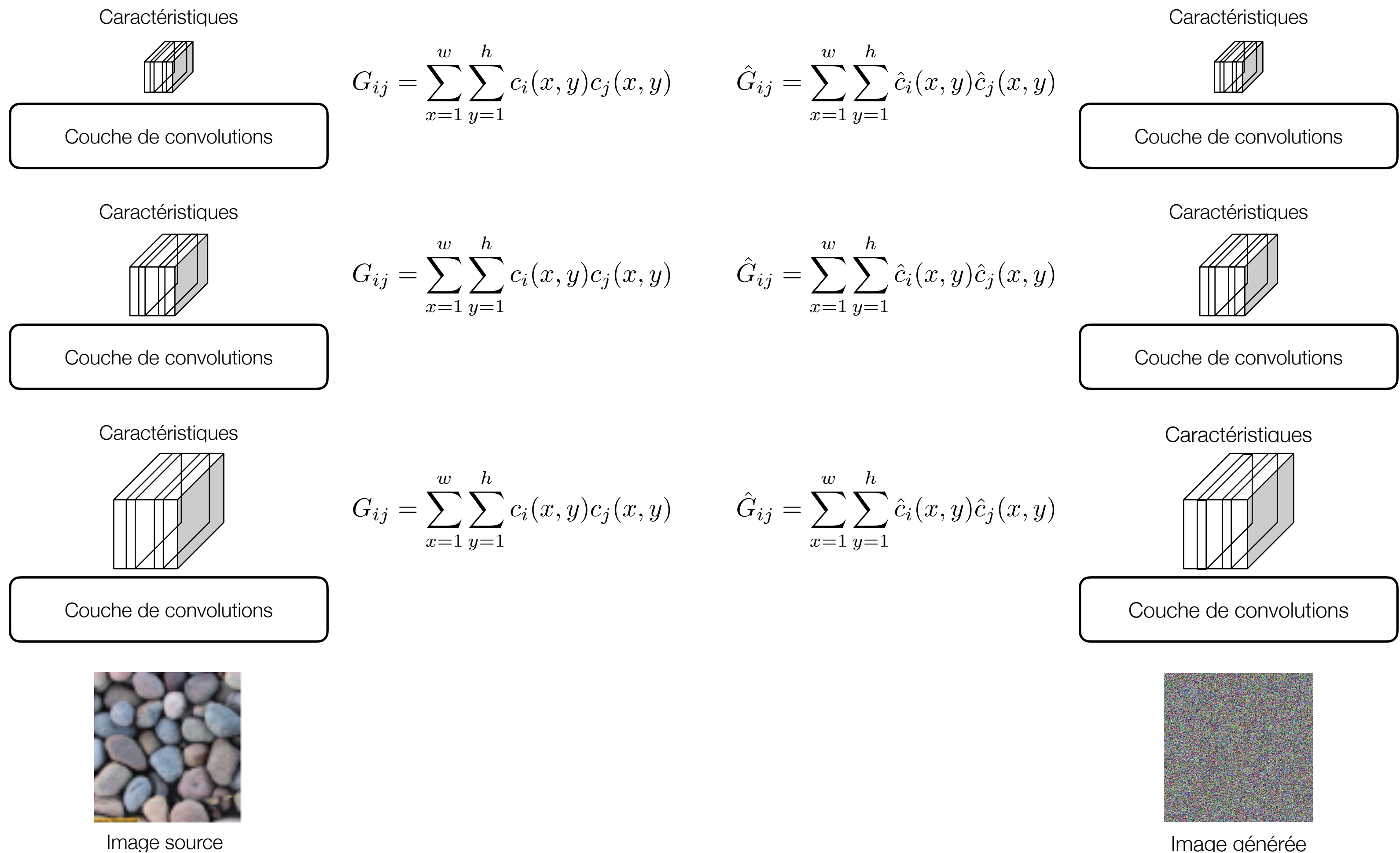
Synthèse de textures par CNN



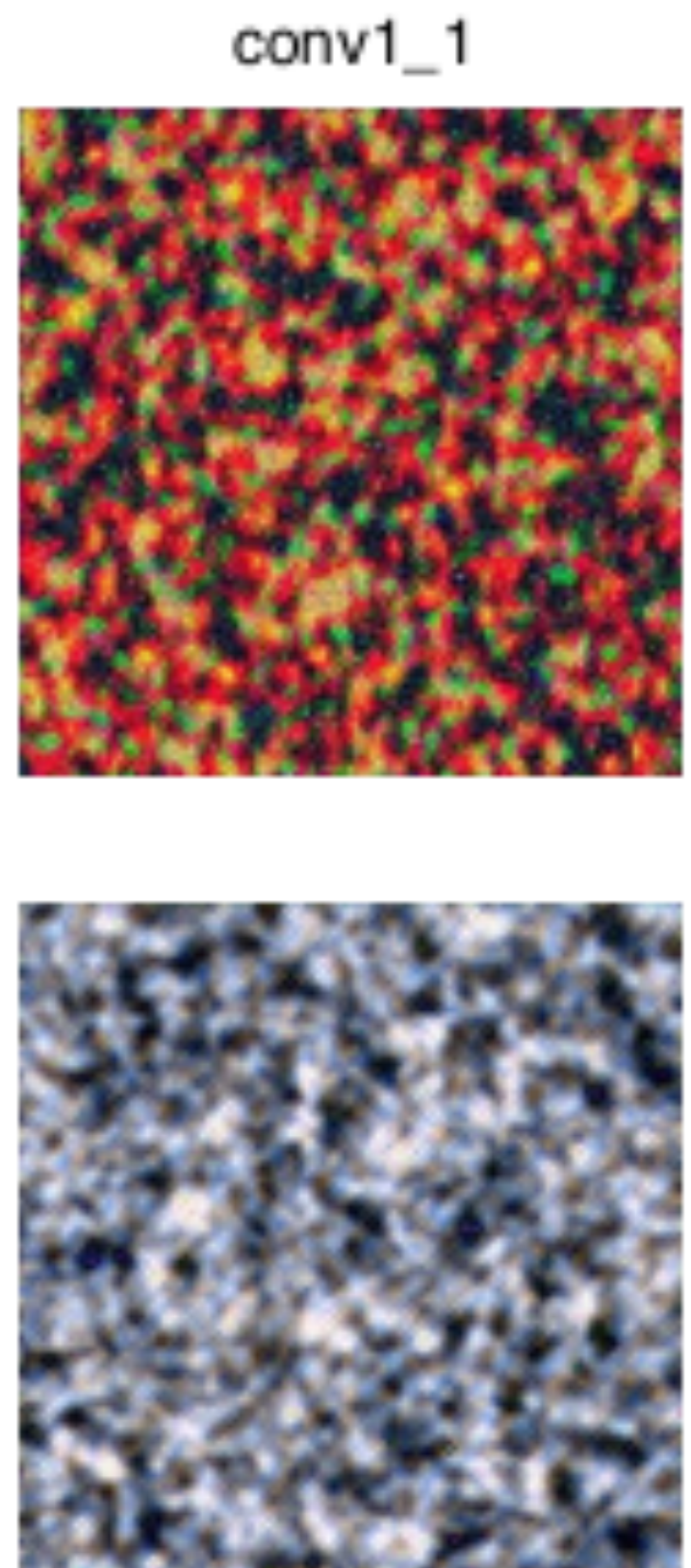
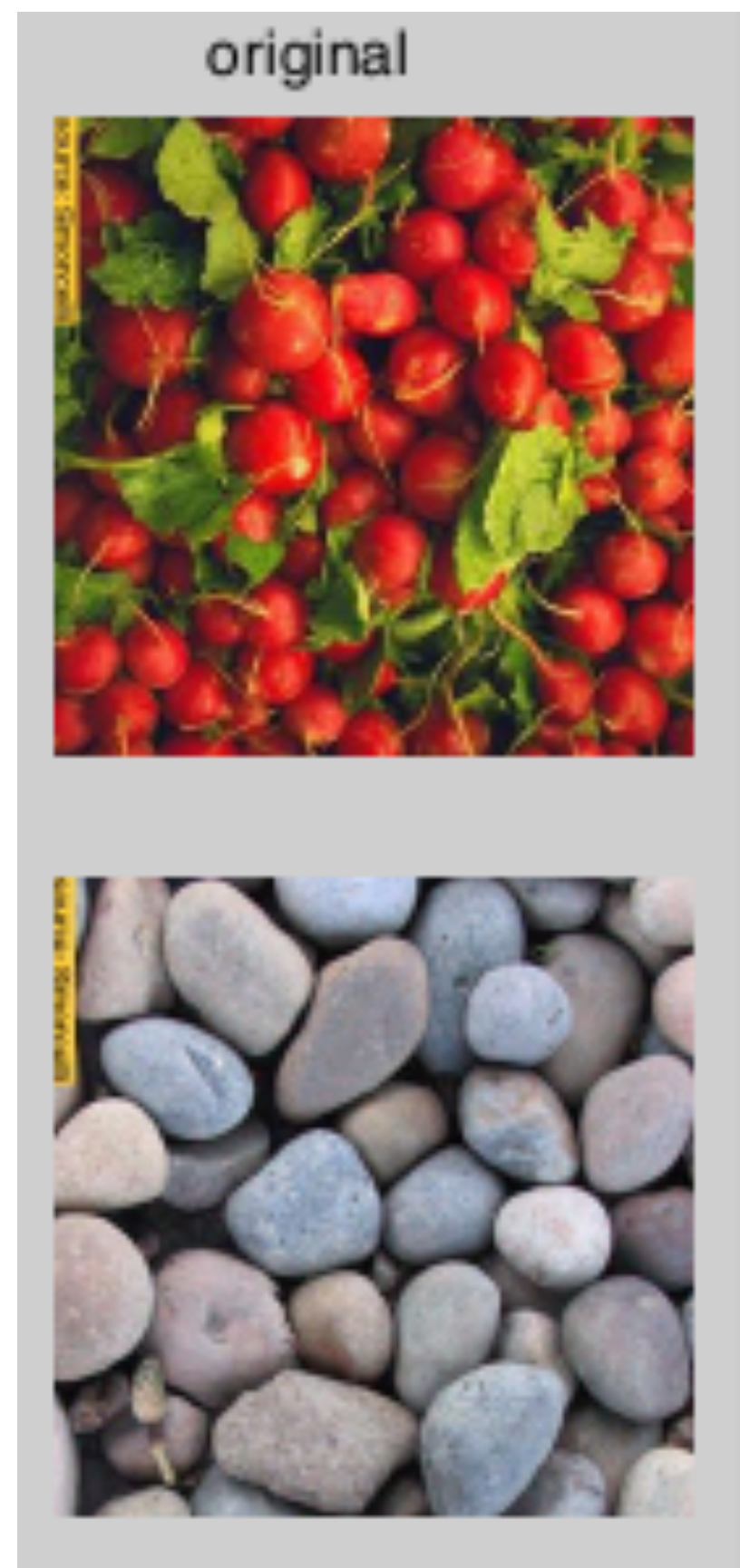
Synthèse de textures par CNN



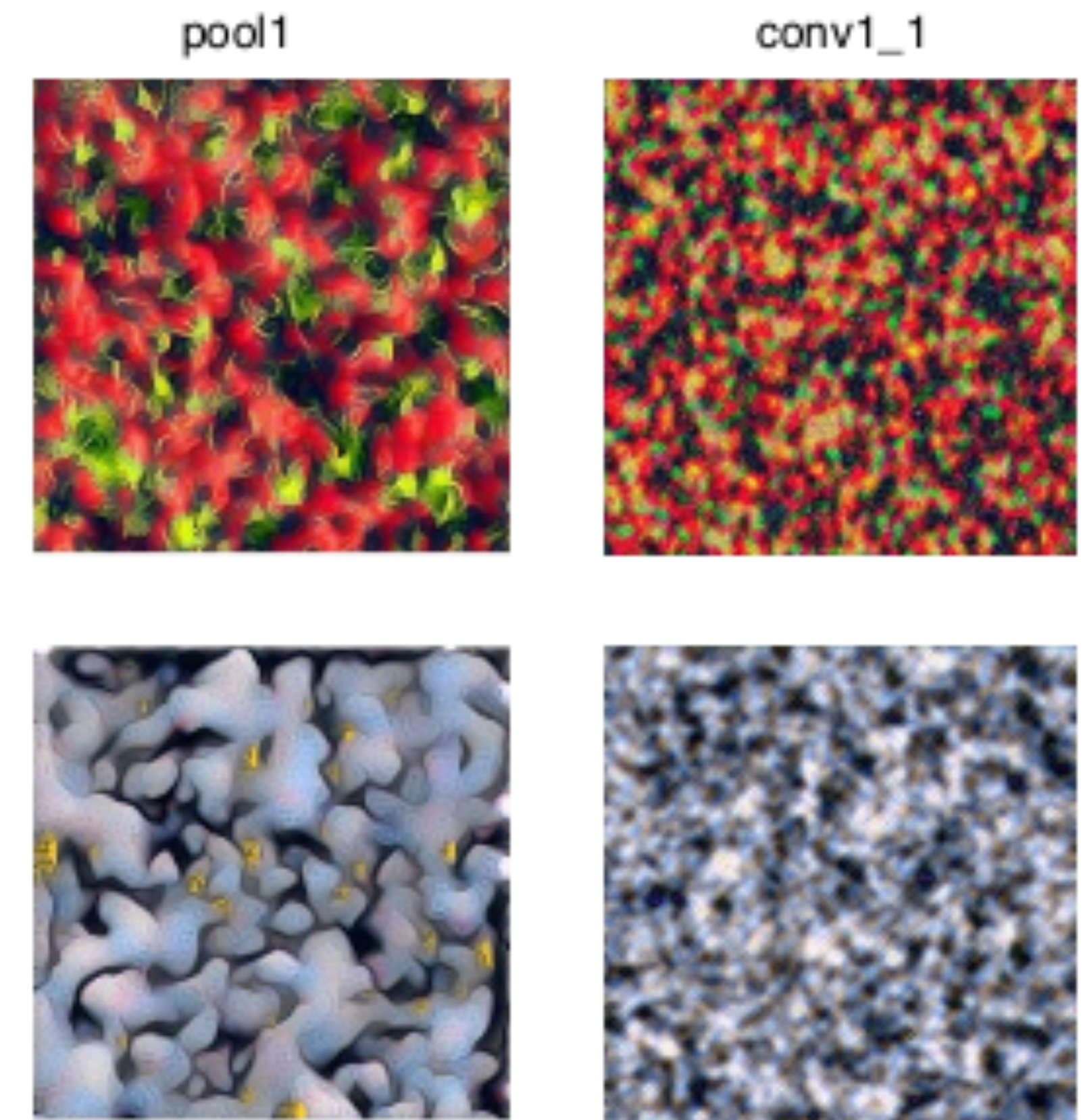
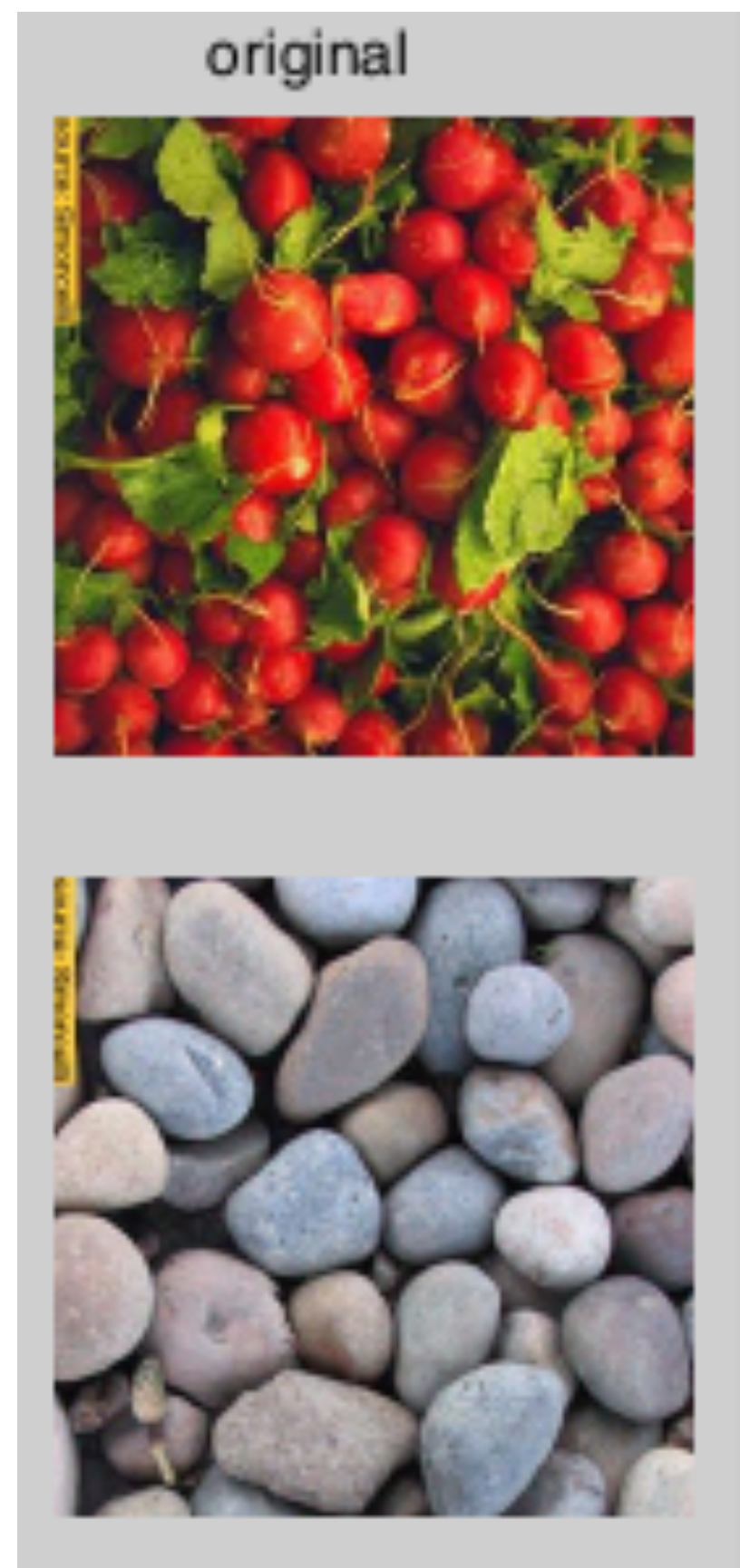
Synthèse de textures par CNN



Synthèse de textures par CNN



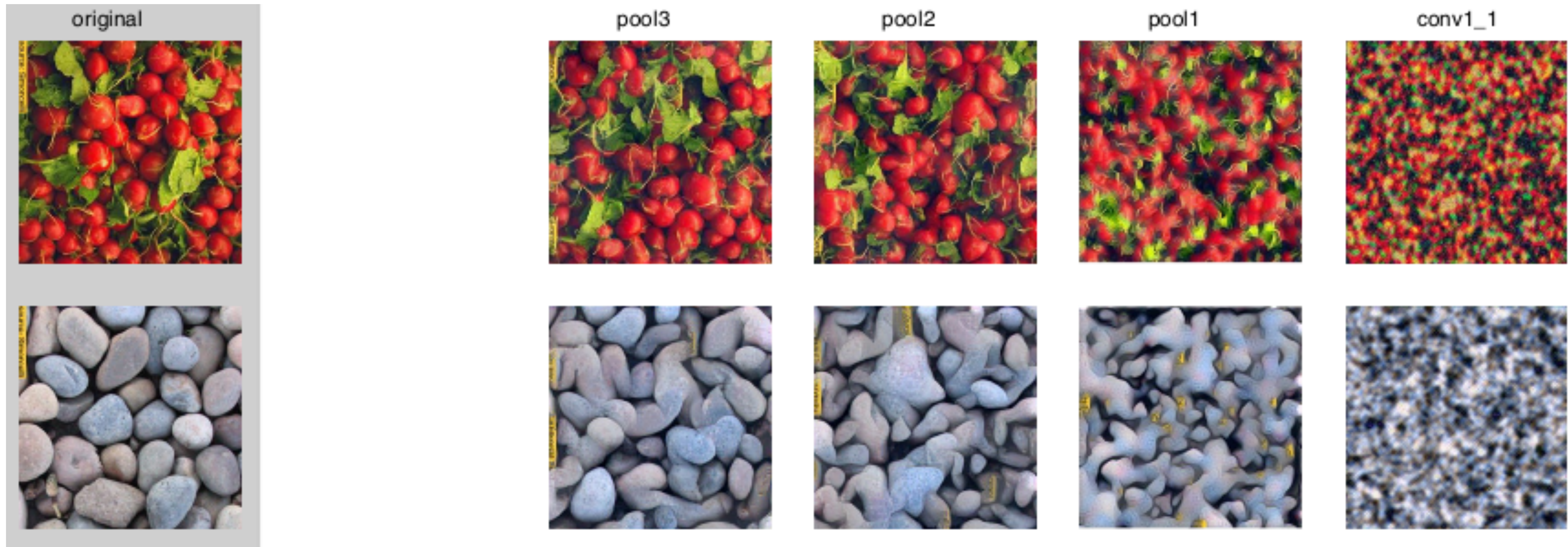
Synthèse de textures par CNN



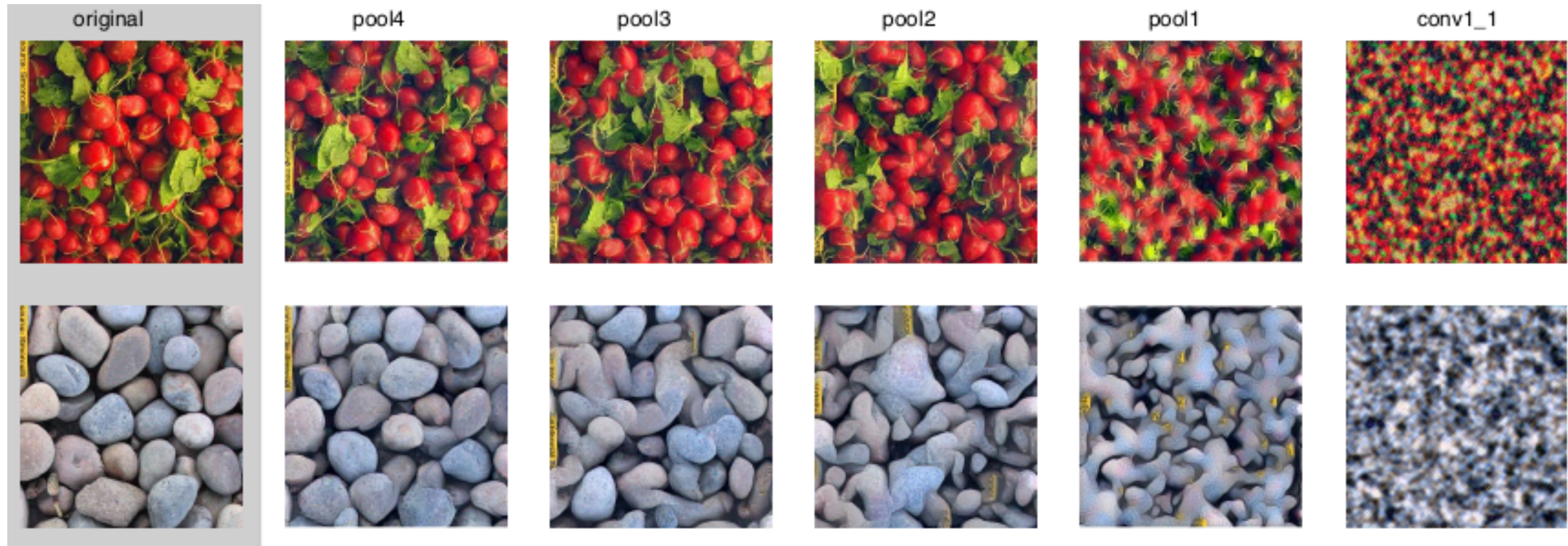
Synthèse de textures par CNN



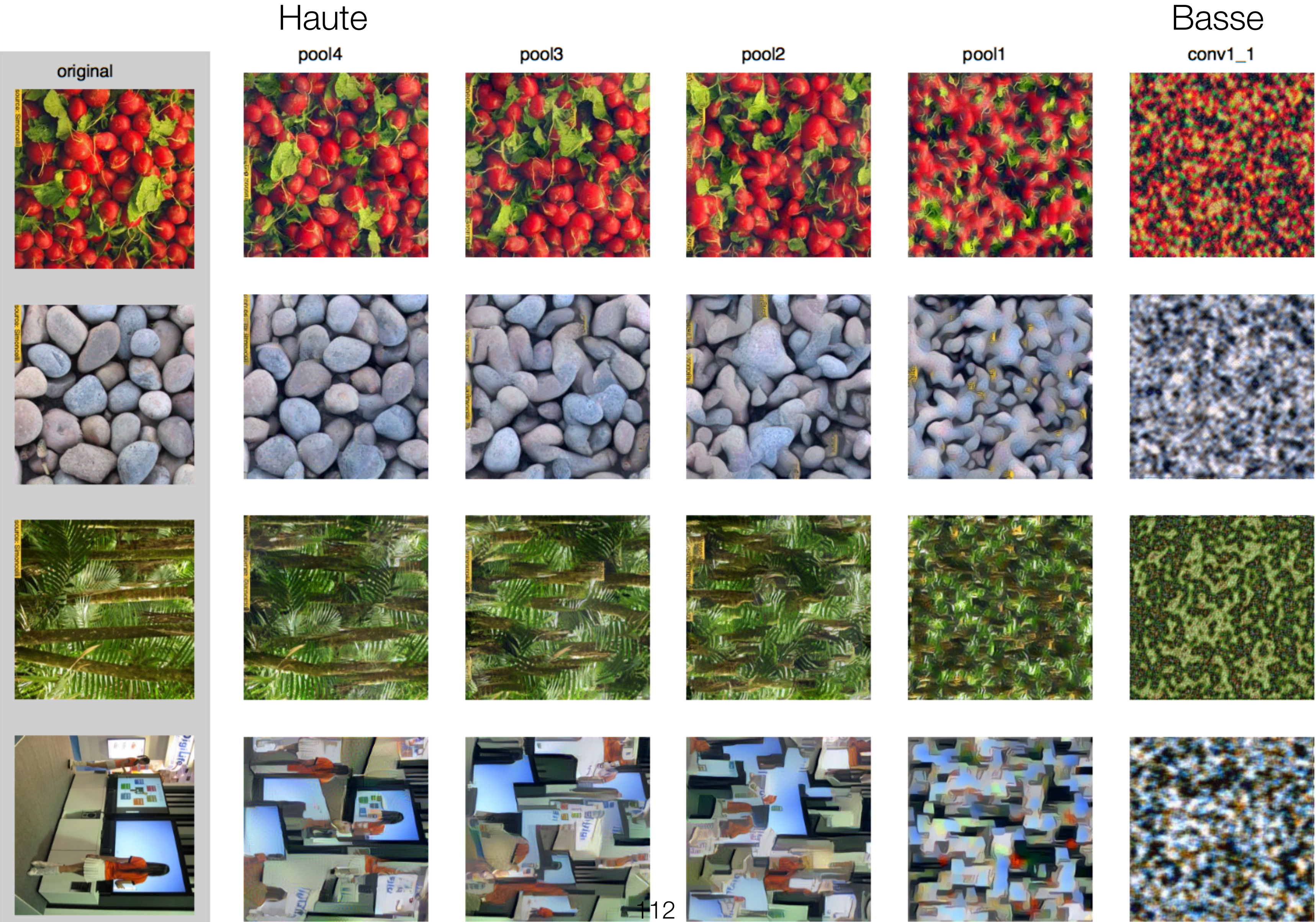
Synthèse de textures par CNN



Synthèse de textures par CNN



Statistiques de textures



Synthèse de textures par CNN

Synthèse



Source



Synthèse de textures par CNN

Synthèse



Source



Génération d'images par apprentissage profond

Transfert de style par appariement de caractéristiques profondes

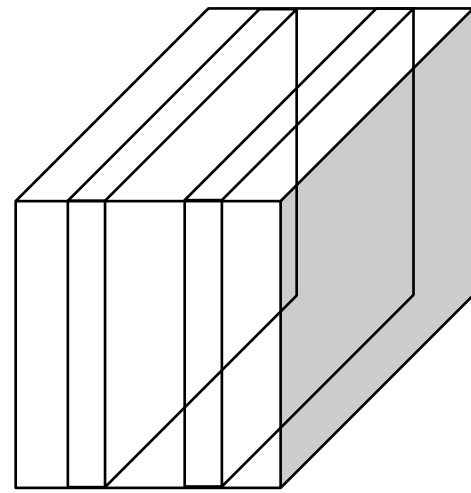


GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

Synthèse de textures par CNN

$$G_{ij} = \sum_{x=1}^w \sum_{y=1}^h c_i(x, y) c_j(x, y)$$

Caractéristiques



Couche de convolutions



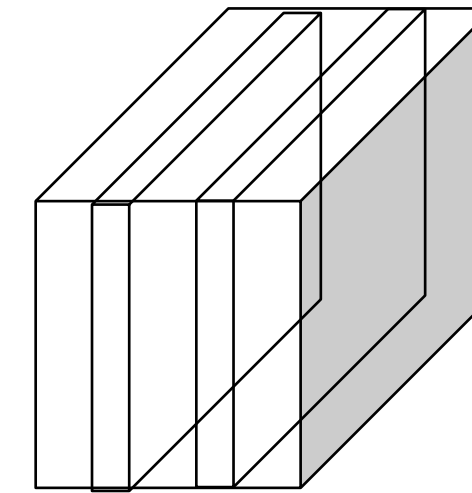
Image source

Minimise la différence entre les corrélations pour reproduire la texture!

$$\sum_{i,j} (G_{ij} - \hat{G}_{ij})^2$$

$$\hat{G}_{ij} = \sum_{x=1}^w \sum_{y=1}^h \hat{c}_i(x, y) \hat{c}_j(x, y)$$

Caractéristiques



Couche de convolutions



Image générée

Capturer le style artistique

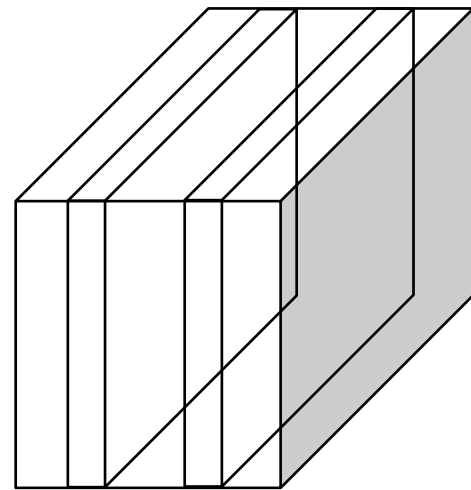
Comment transférer le style d'une peinture vers une photo?



Transfert de style

$$G_{ij} = \sum_{x=1}^w \sum_{y=1}^h c_i(x, y) c_j(x, y)$$

Caractéristiques



Couche de convolutions



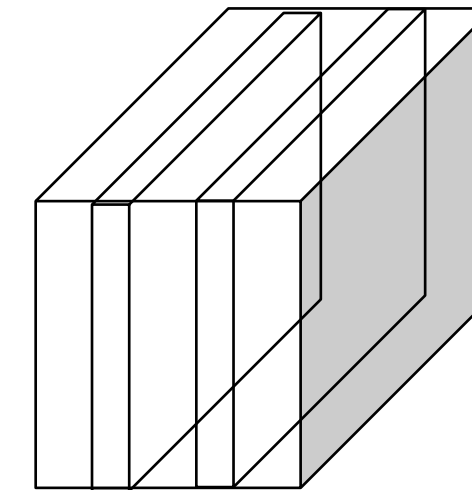
Image de style

Minimise la différence entre les corrélations pour reproduire la texture (**style**)

$$\sum_{i,j} (G_{ij} - \hat{G}_{ij})^2$$

$$\hat{G}_{ij} = \sum_{x=1}^w \sum_{y=1}^h \hat{c}_i(x, y) \hat{c}_j(x, y)$$

Caractéristiques



Couche de convolutions

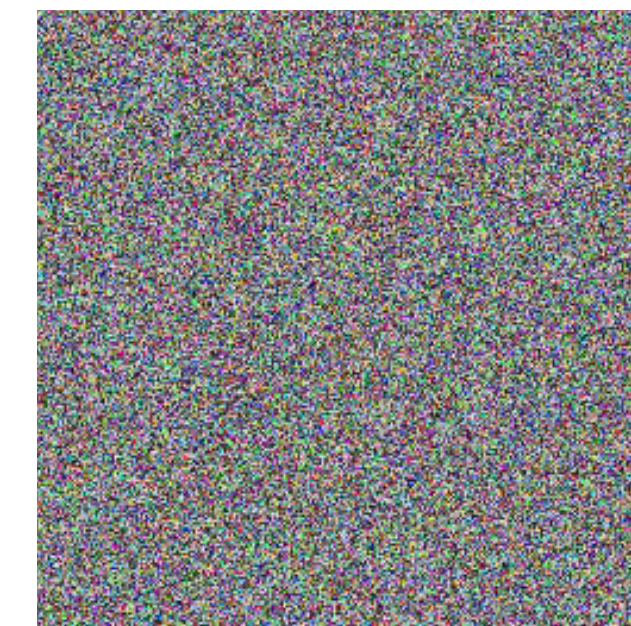
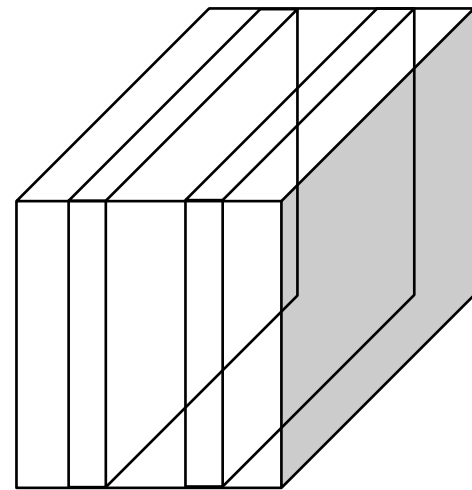


Image générée

Transfert de style

$$G_{ij} = \sum_{x=1}^w \sum_{y=1}^h c_i(x, y) c_j(x, y)$$

Caractéristiques



Couche de convolutions



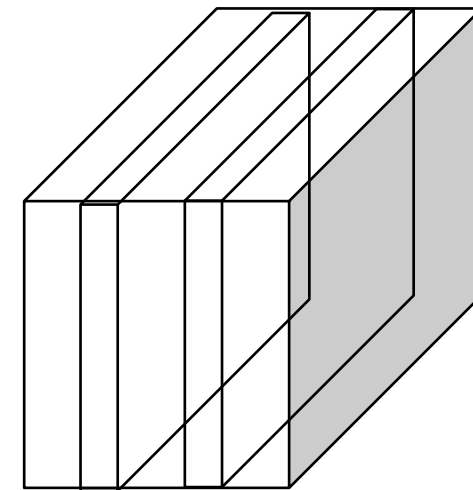
Image de style

$$\sum_{i,j} (G_{ij} - \hat{G}_{ij})^2$$

Minimise la différence entre les corrélations pour reproduire la texture (**style**)

$$\hat{G}_{ij} = \sum_{x=1}^w \sum_{y=1}^h \hat{c}_i(x, y) \hat{c}_j(x, y)$$

Caractéristiques



Couche de convolutions

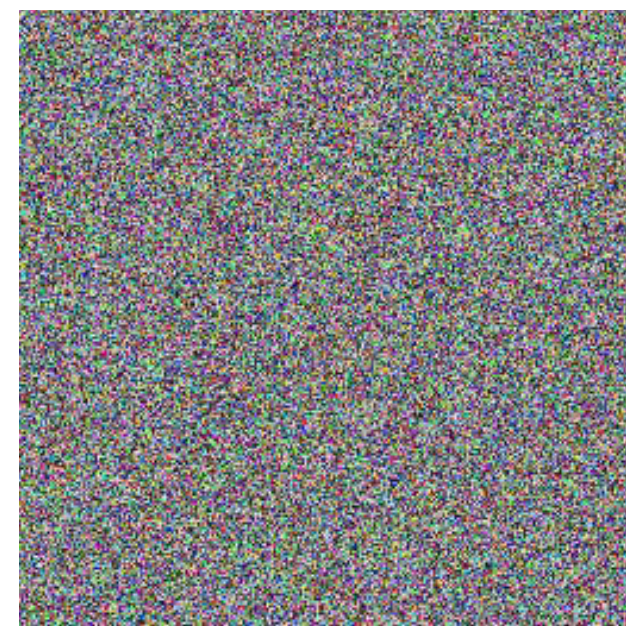
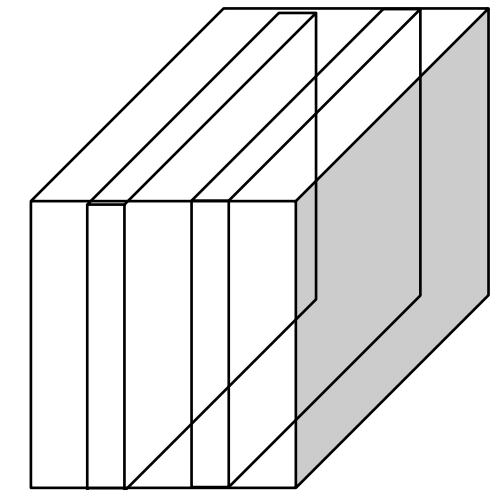


Image générée

$$\sum_i \sum_{x,y} (c_i(x, y) - \hat{c}_i(x, y))^2$$

Caractéristiques



Couche de convolutions



Image de contenu











Londres le jour



New York le soir

Londres le soir?



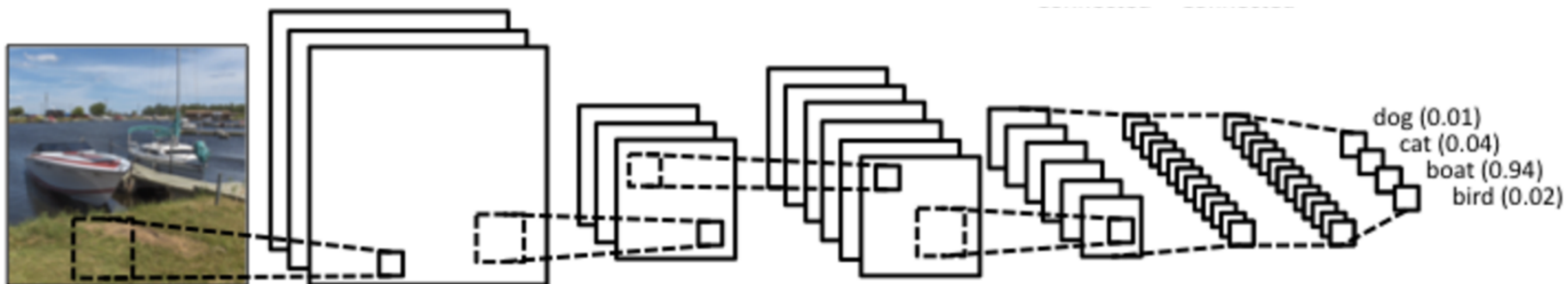
Génération d'images par apprentissage profond

Colorisation d'image par réseaux profonds

GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

L'architecture d'un réseau à convolutions

Image en entrée



L'architecture d'un réseau à convolutions discriminatif

Image en entrée



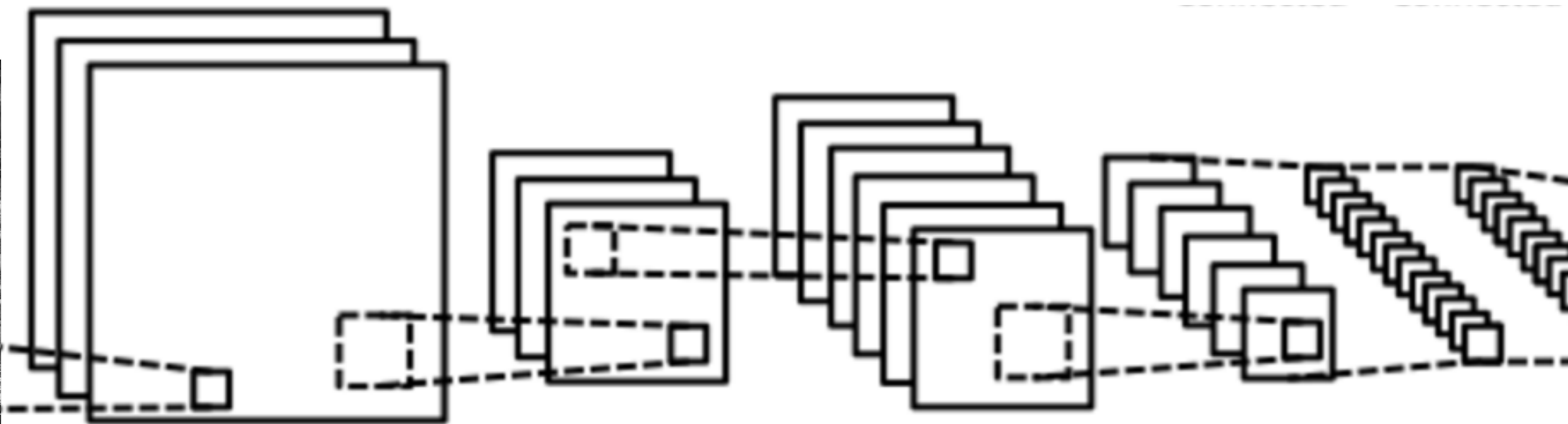
L'architecture d'un réseau à convolutions génératif

Rappel

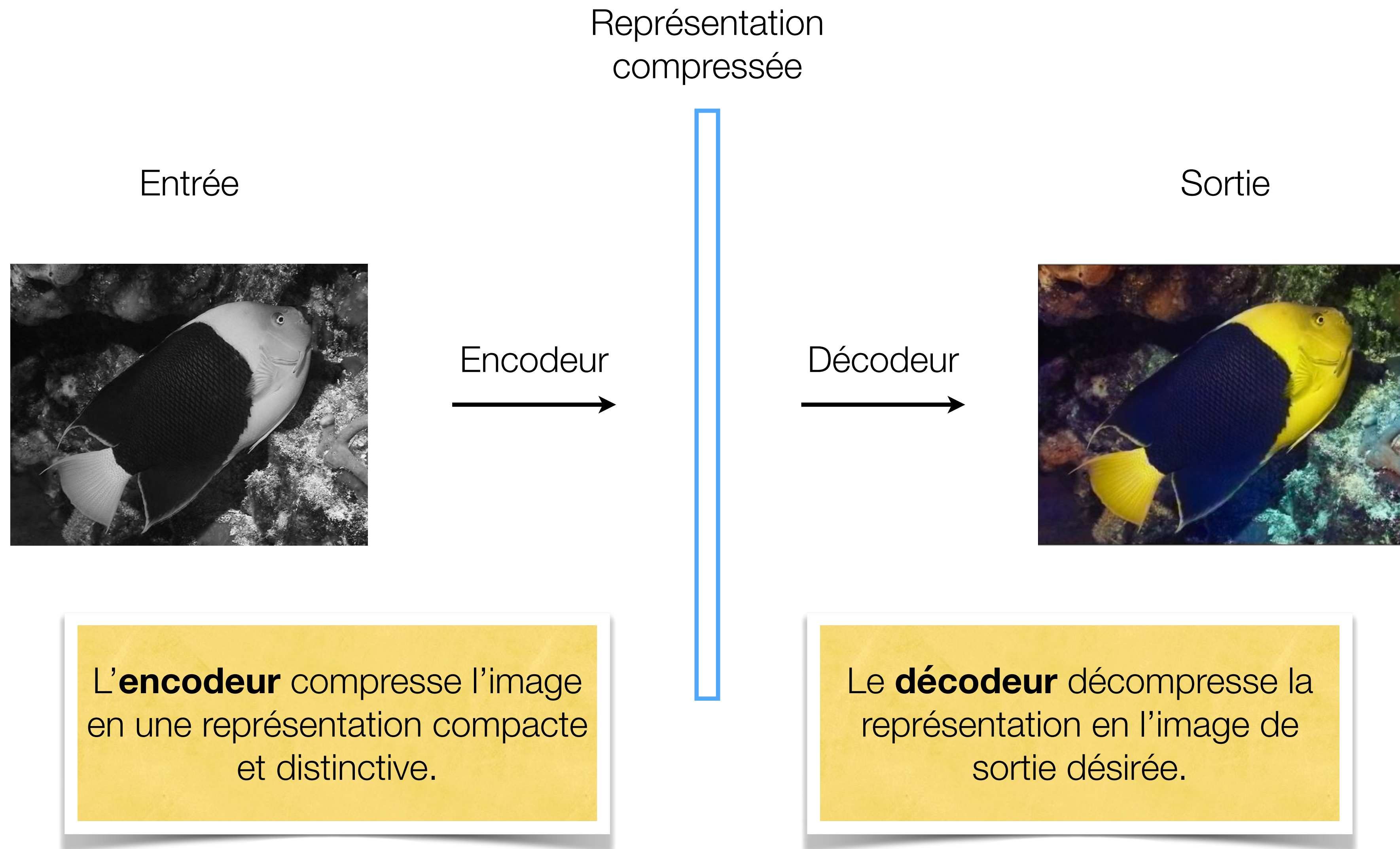
Image en entrée



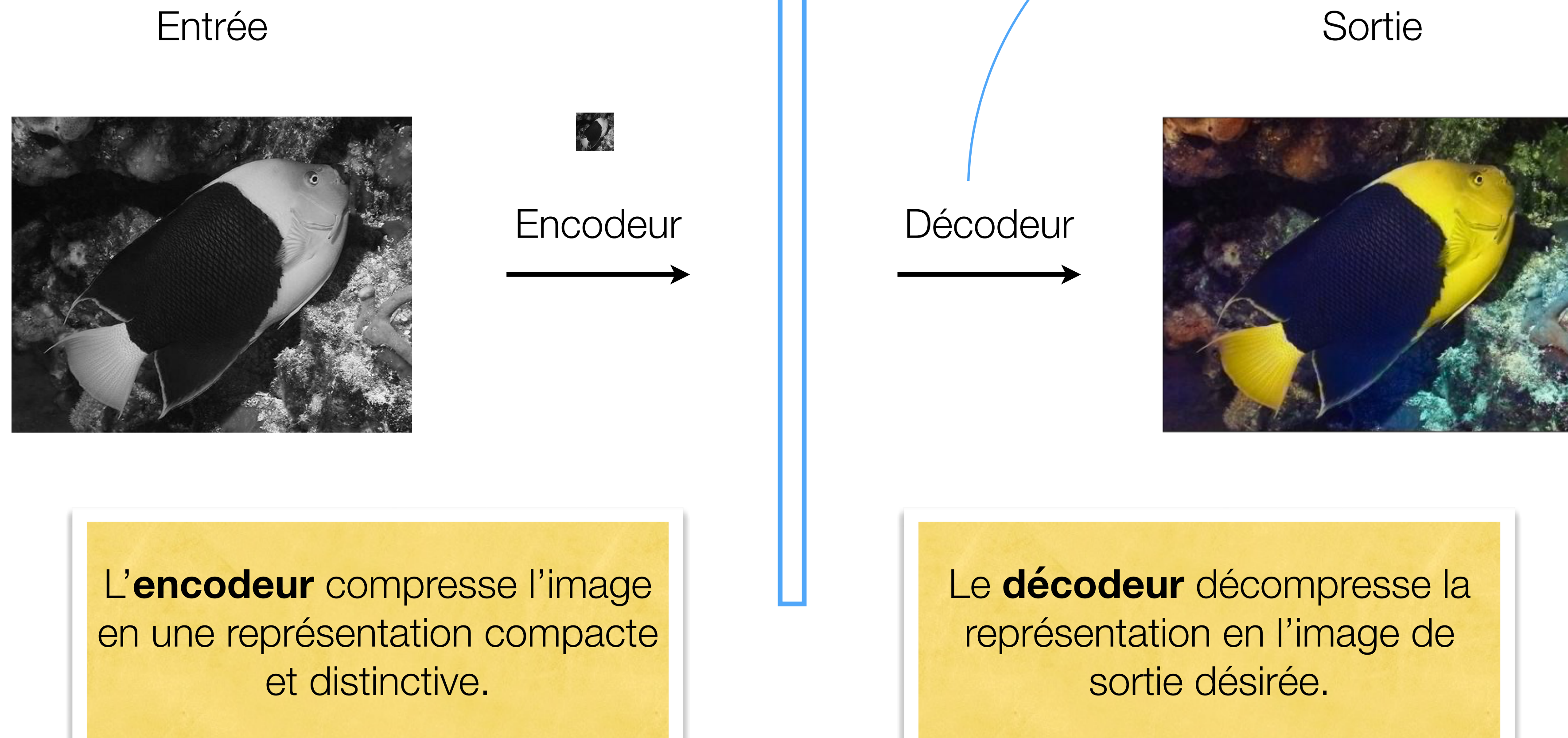
Image en sortie?



Autre point de vue : encodeur/décodeur



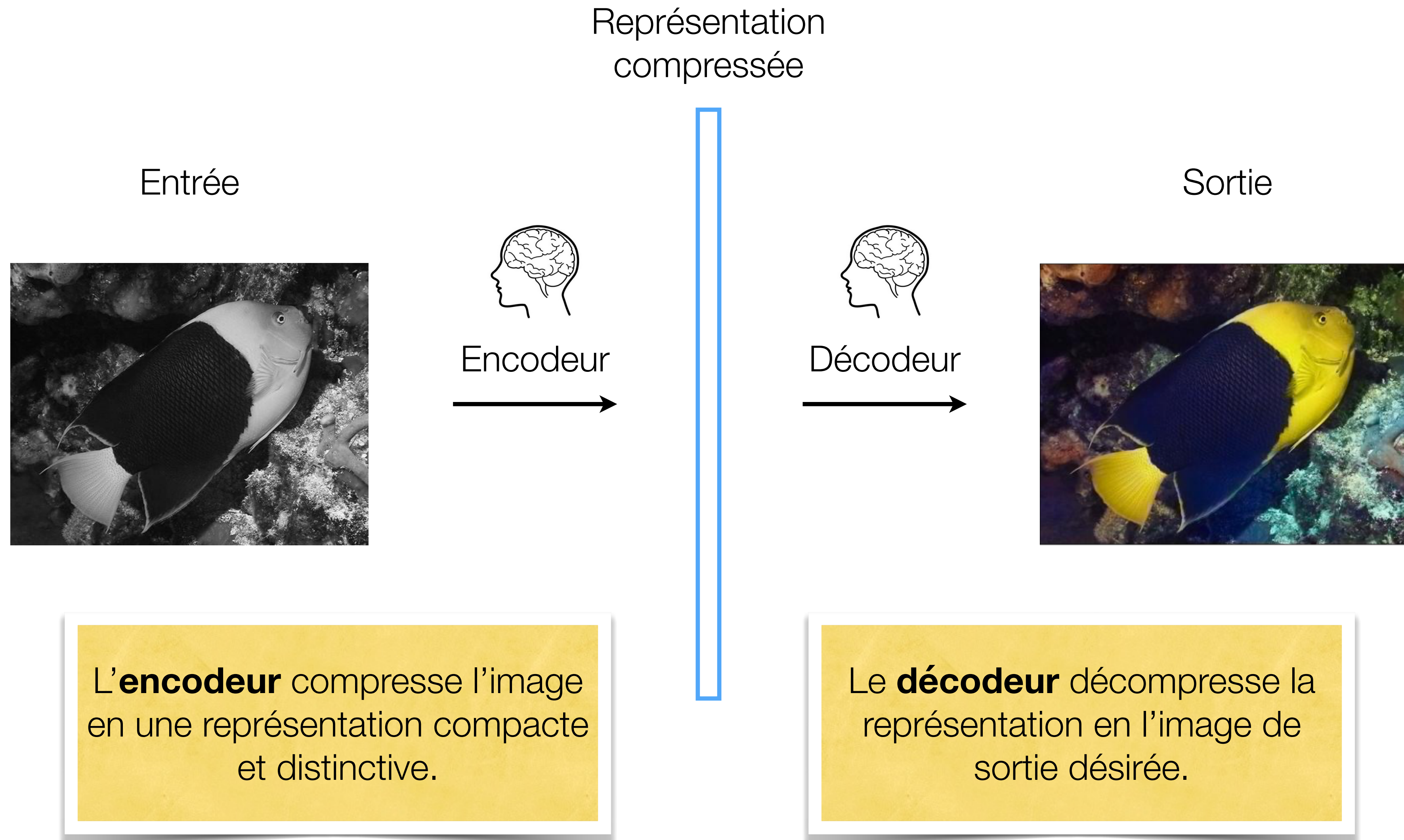
Encodeur/décodeur



Rappel

Encodeur/décodeur

Apprendre l'encodeur et le décodeur
(à partir des données)
afin de satisfaire un objectif, ici : colorer l'image.

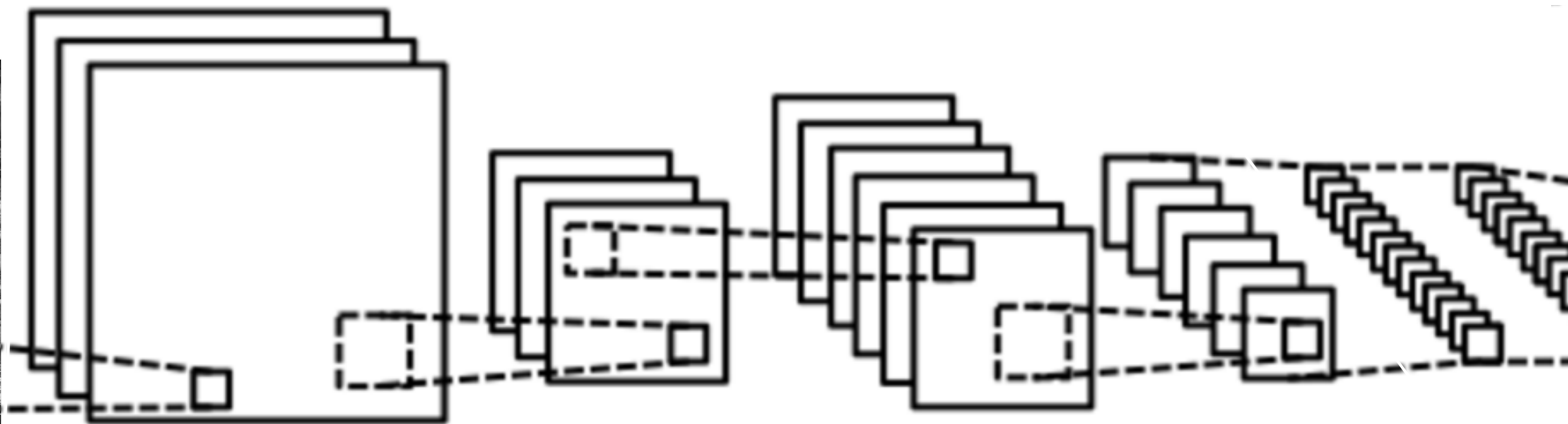


L'architecture d'un réseau à convolutions génératif

Image en entrée



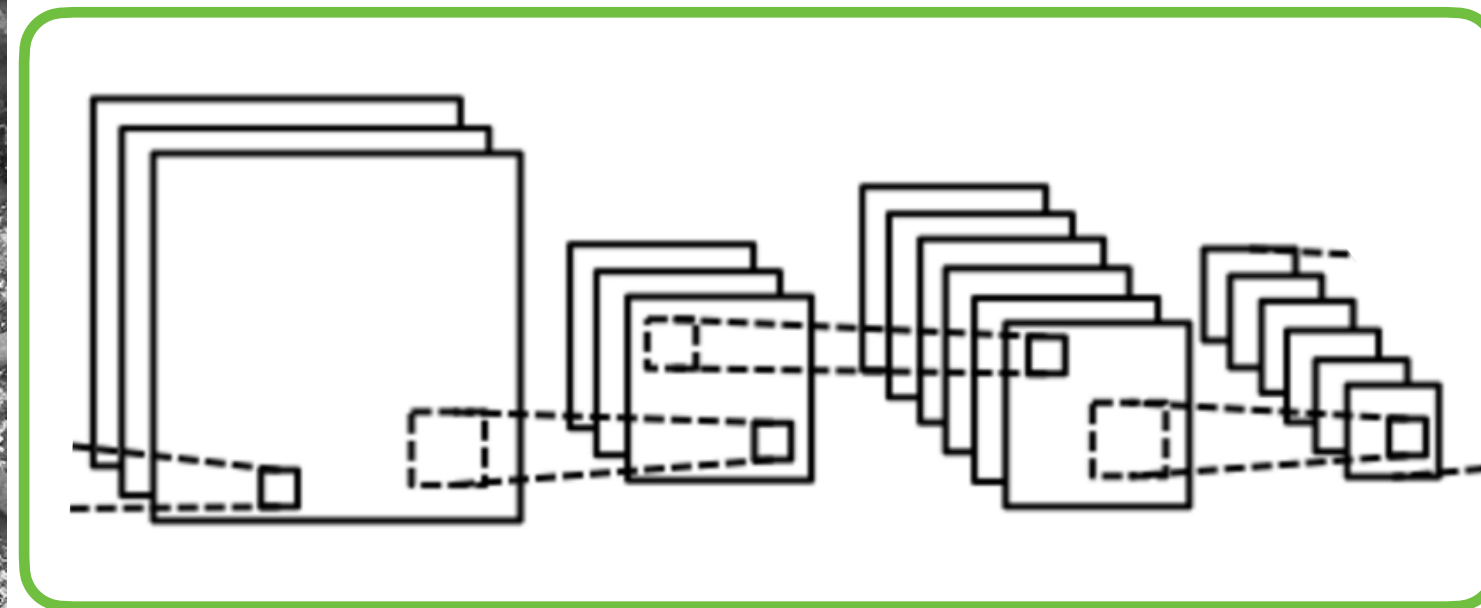
Image en sortie?



L'architecture d'un réseau à convolutions génératif

Représentation
compressée

Image en entrée



Couches de convolution

Comprime l'image en une représentation apprise

Image en sortie?



Couches de « dé-convolution »

Dé-comprime la représentation apprise en une autre image

L'architecture d'un réseau à convolutions génératif

Représentation
compressée

Image en sortie?

Utiliser le terme « dé-convolution »
est un **abus de langage**!
Il s'agit simplement de convolutions
entrecoupées de redimensionnement des
caractéristiques.

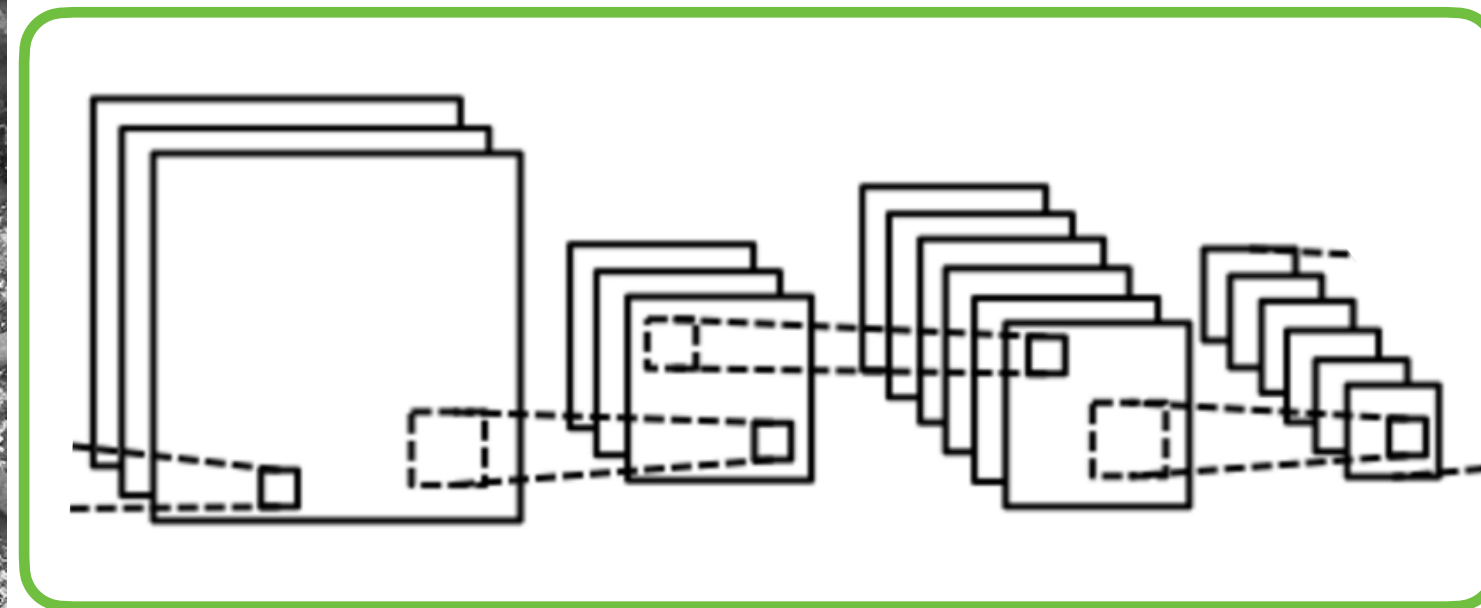


Couches de « dé-convolution »
Dé-compresse la représentation
apprise en une autre image

L'architecture d'un réseau à convolutions génératif

Représentation
compressée

Image en entrée



Couches de convolution

Comprime l'image en une
représentation apprise



Couches de « dé-convolution »

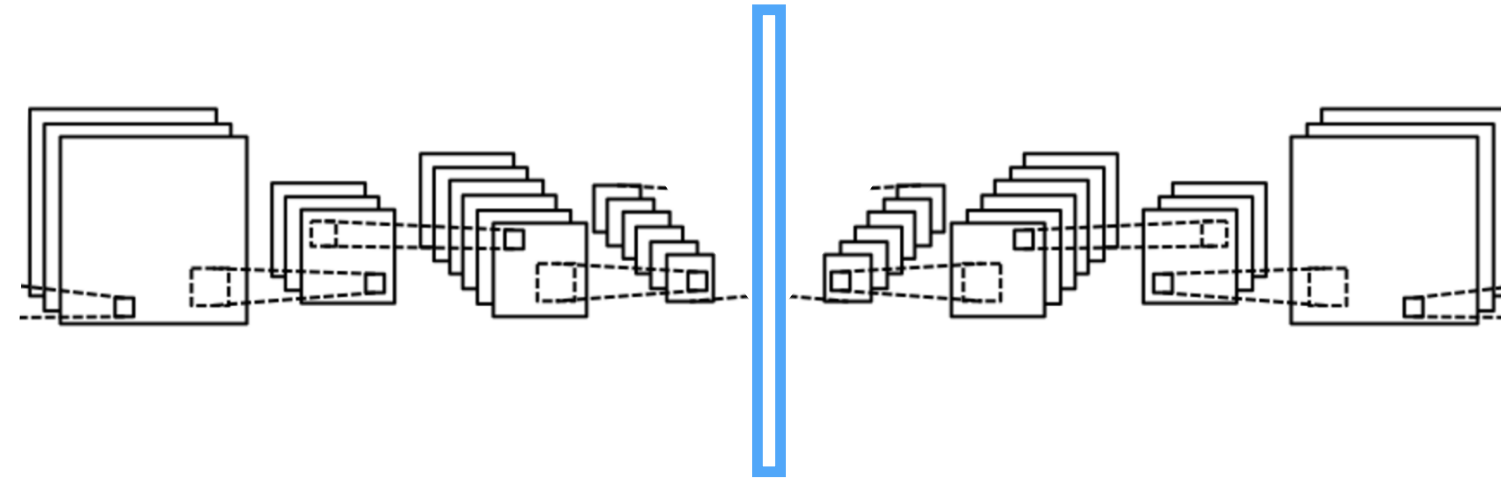
Dé-comprime la représentation
apprise en une autre image

Image en sortie?





Image d'entrée : canal L (de Lab)



Sortie : canaux ab (de Lab)

Malheureusement, minimiser la somme
des différences au carré
ne fonctionne pas!

$$L_2(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \hat{\mathbf{Y}}_{h,w}\|_2^2$$

Entrée



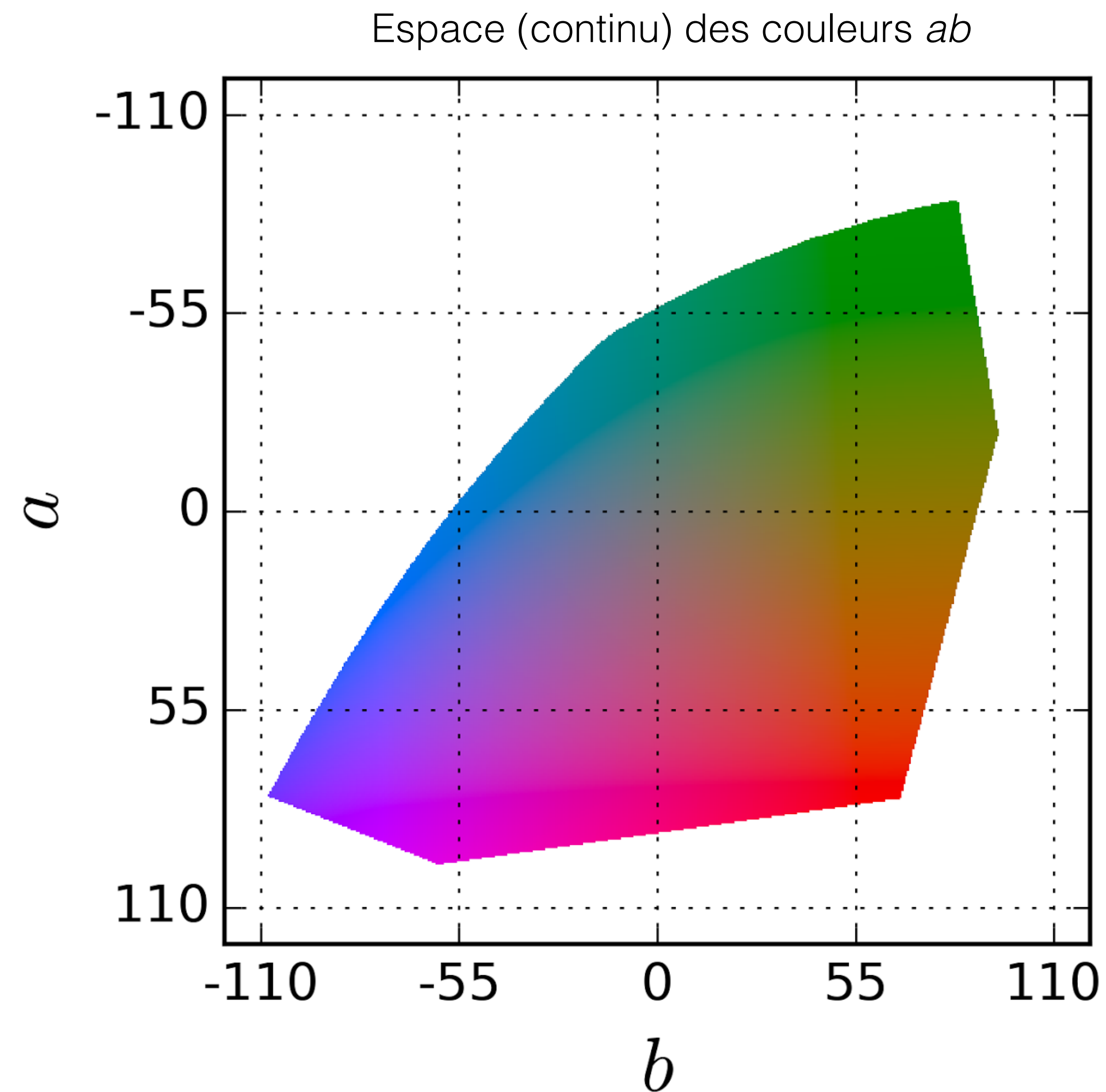
Sortie



Vérité

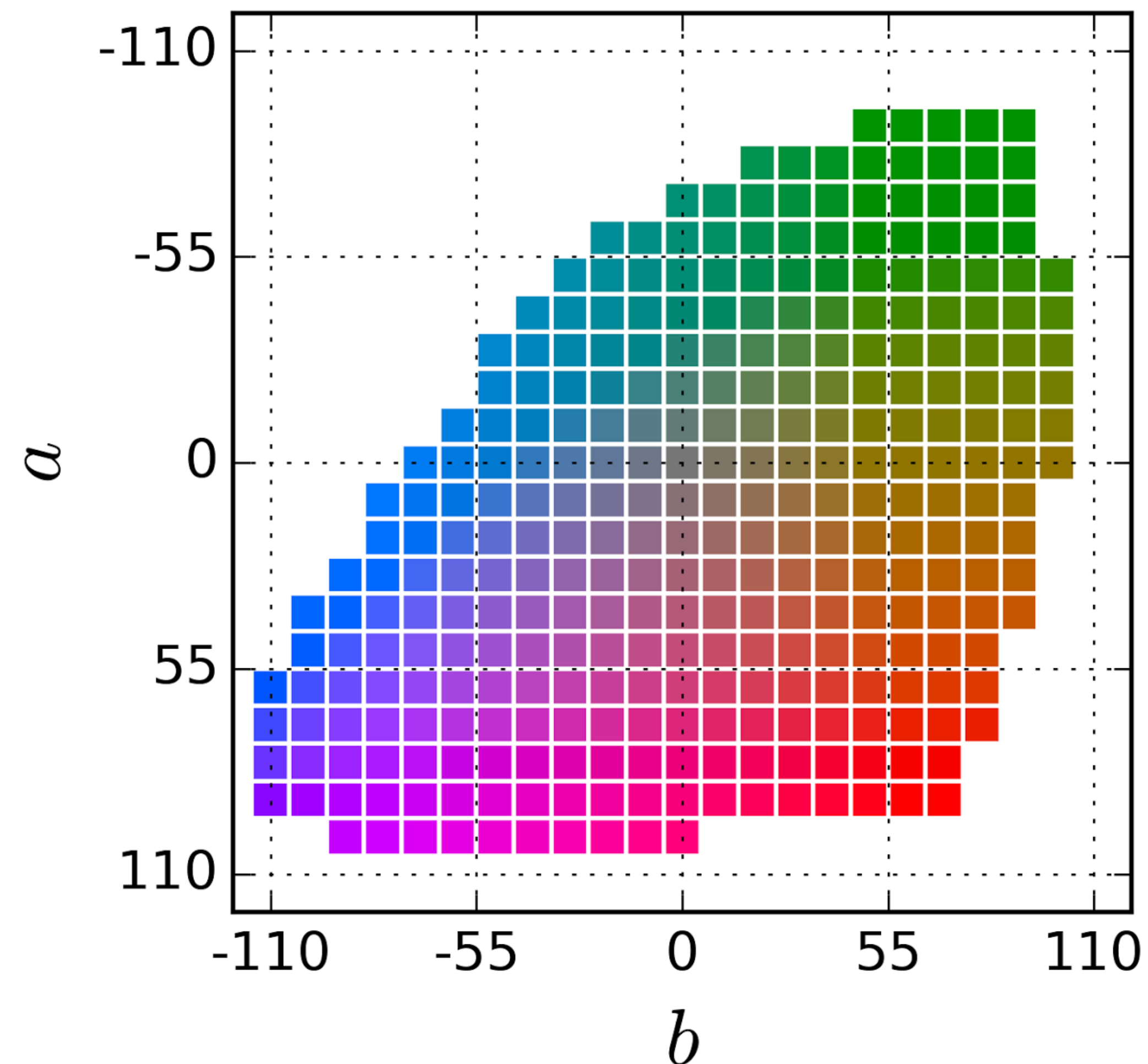


Solution : meilleure fonction de perte



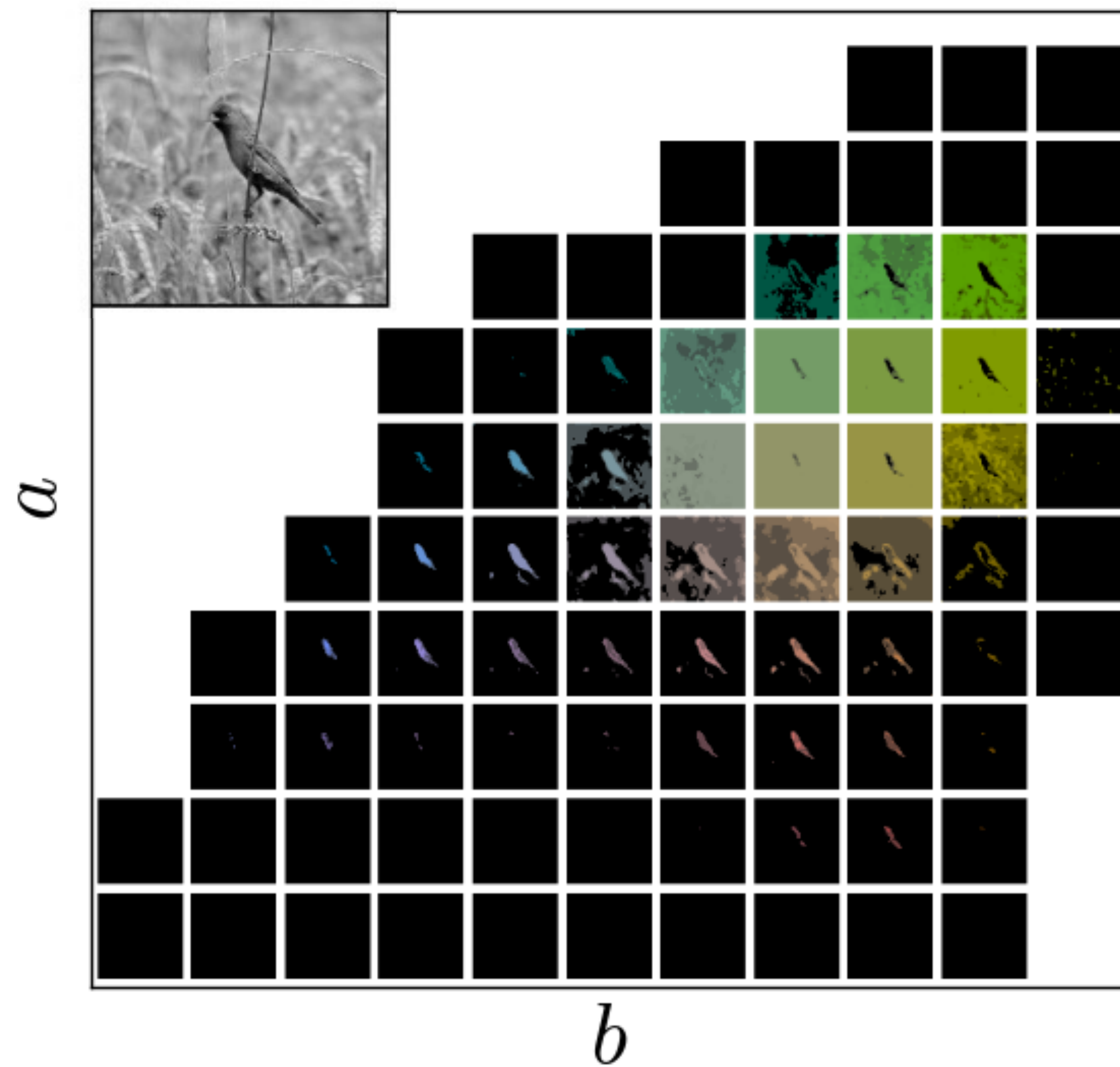
Solution : meilleure fonction de perte

Discrétisation de l'espace de couleurs ab (313 bins)



Prédire la distribution sur
l'espace ab !

Permet de modéliser les
distributions multimodales
(plusieurs couleurs sont
plausibles)



Couleur à partir de la distribution

$$f_T(\mathbf{z}) = \frac{\exp(\log(\mathbf{z})/T)}{\sum_q \exp(\log(\mathbf{z}_q)/T)}$$

Mean
T=1



Couleur à partir de la distribution

Mode
 $T \rightarrow 0$



Couleur à partir de la distribution

Mean

$T=1$



$T=.77$



$T=.58$



Annealed-Mean

$T=.38$



$T=.29$



$T=.14$

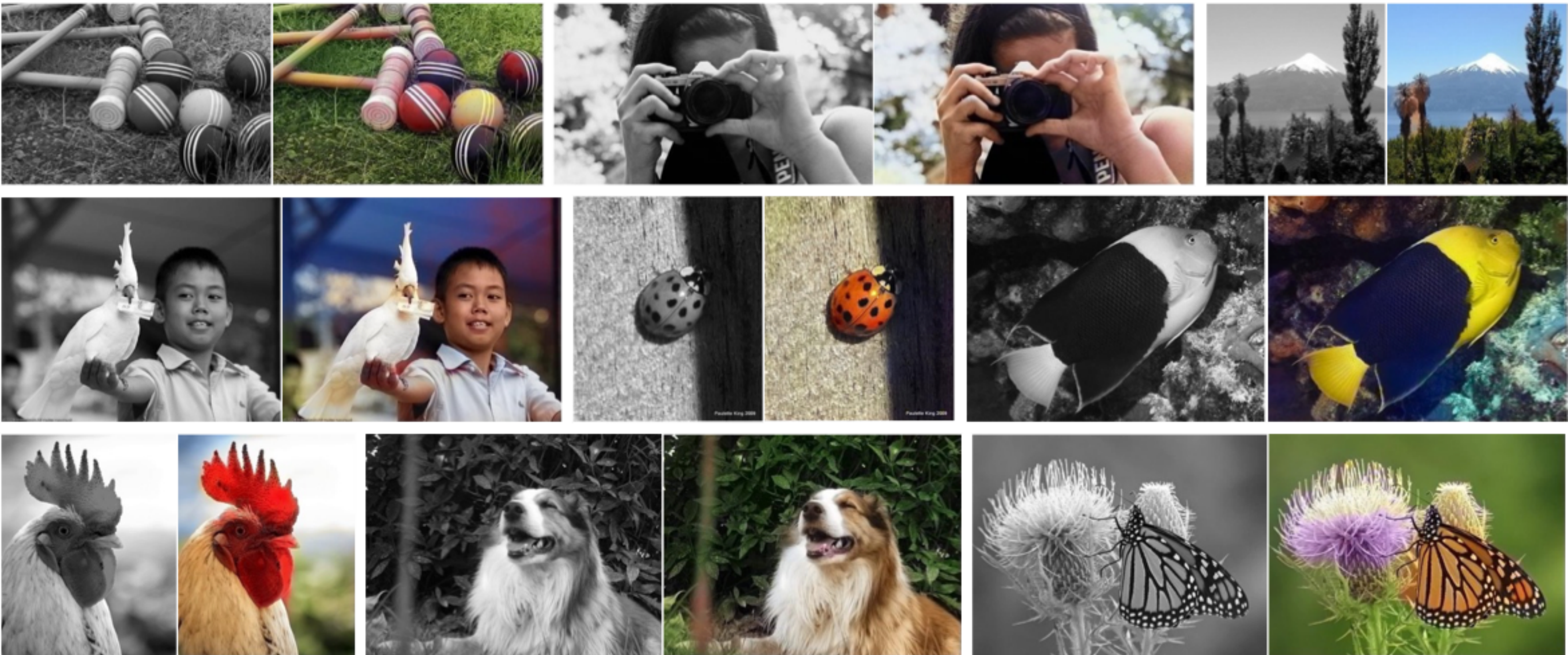


Mode

$T \rightarrow 0$



Bons résultats



Moins bons résultats





Reddit /u/SherySantucci
148





Photo de Reddit /u/Timteroo,
Murale de Eduardo Kobra



Reddit ColorizeBot

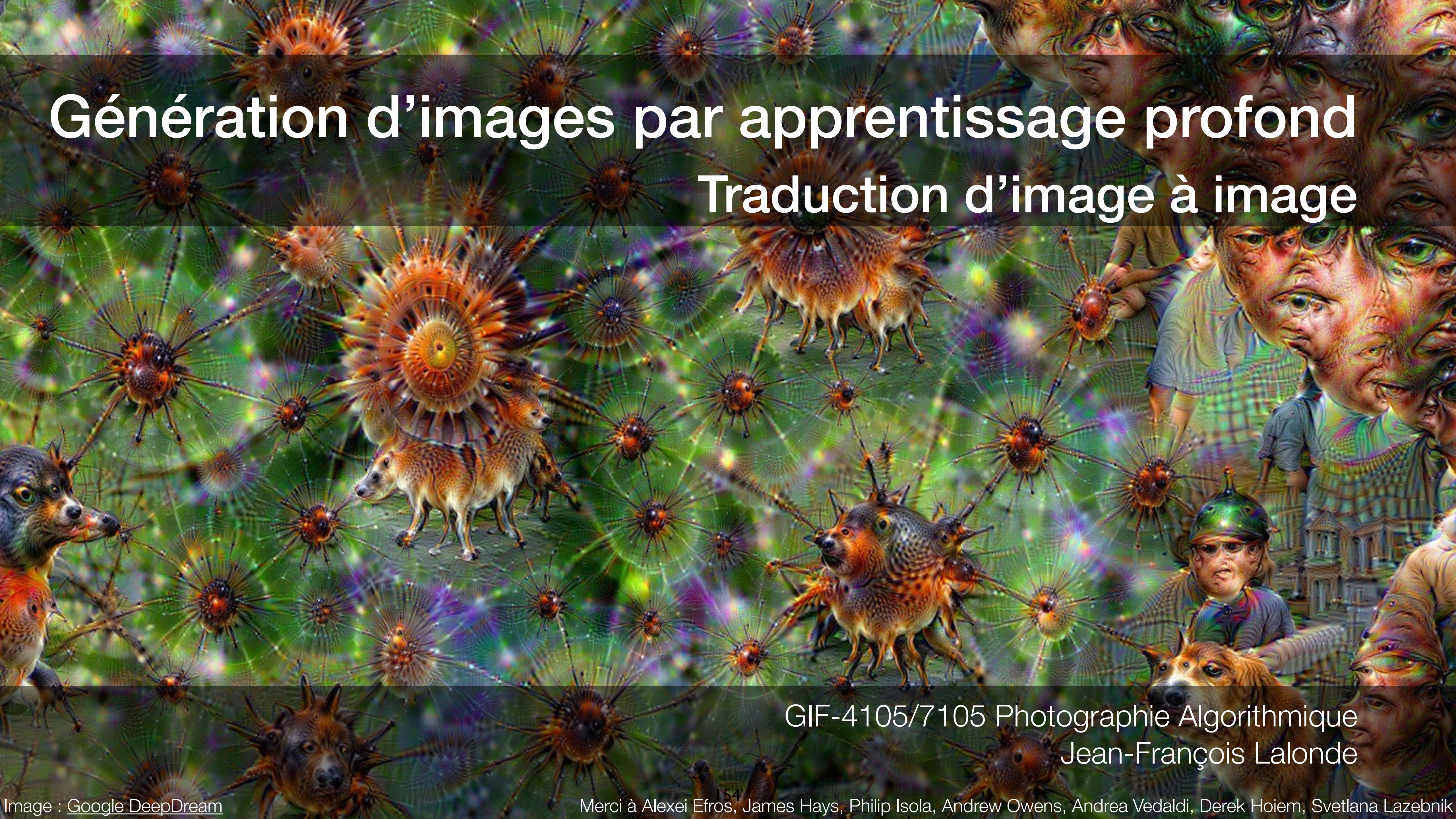


A. Adams, Yosemite Valley Bridge



Génération d'images par apprentissage profond

Traduction d'image à image



GIF-4105/7105 Photographie Algorithmique
Jean-François Lalonde

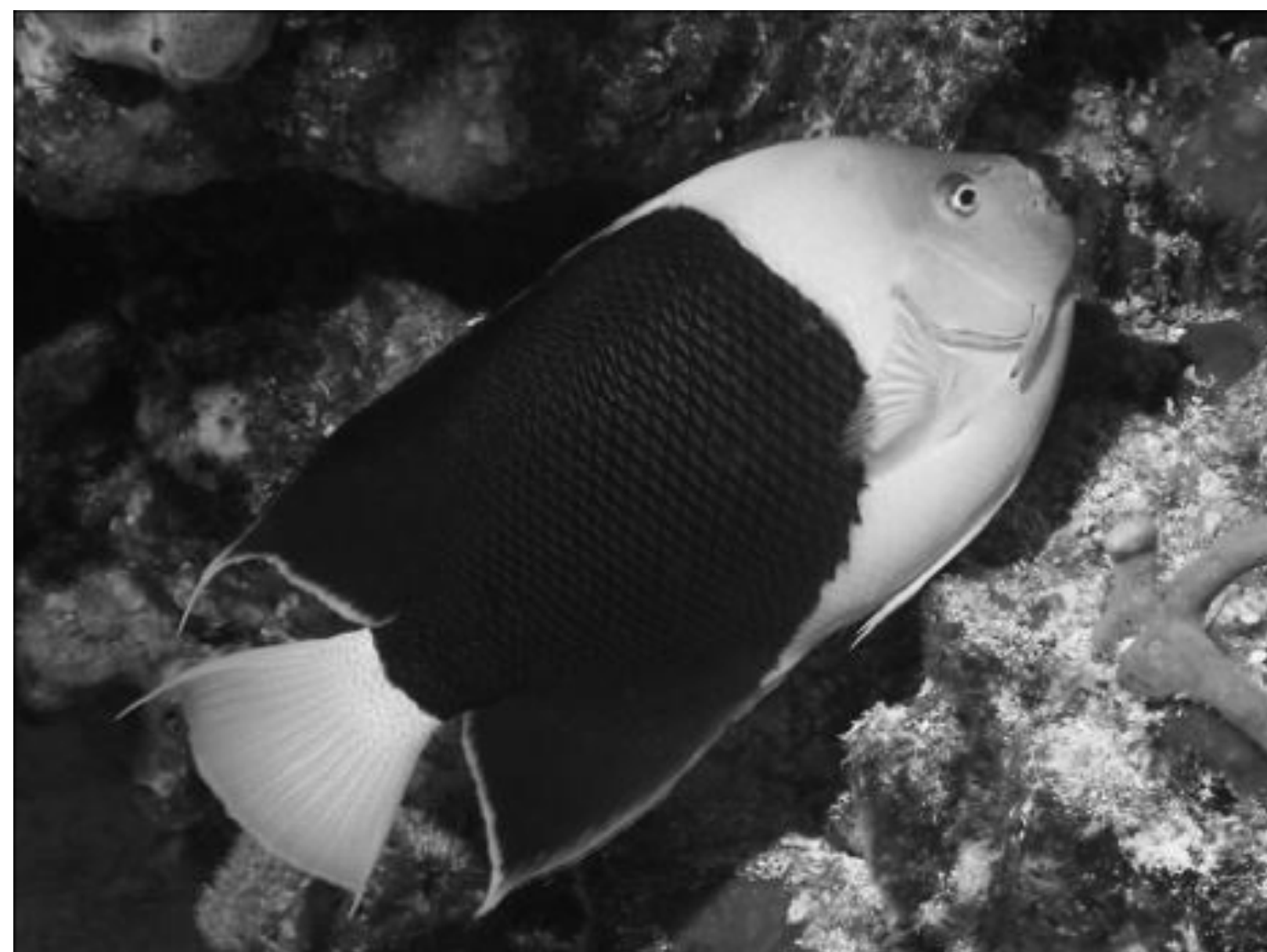
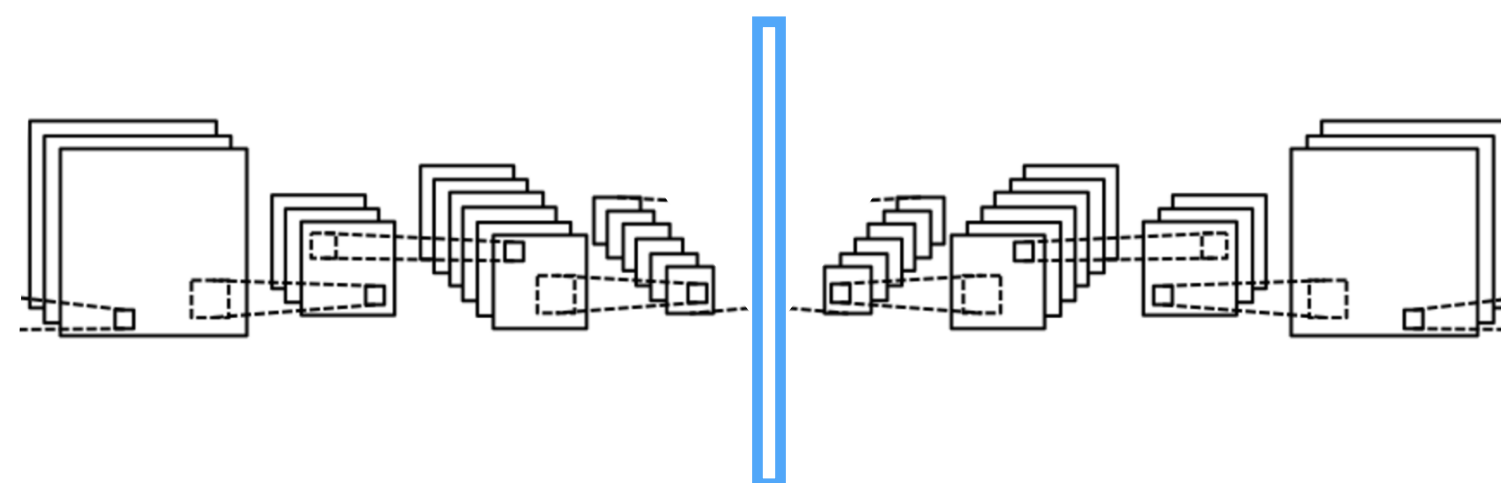


Image d'entrée : canal L (de Lab)



Sortie : canaux ab (de Lab)

Malheureusement, minimiser la somme des différences au carré ne fonctionne pas!

$$L_2(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \hat{\mathbf{Y}}_{h,w}\|_2^2$$

Entrée



Sortie



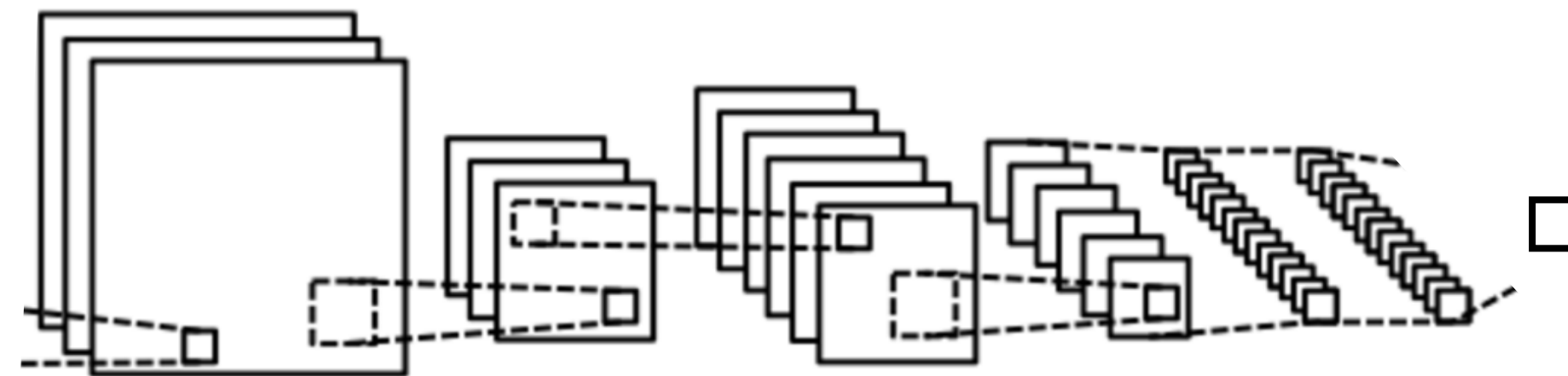
Vérité





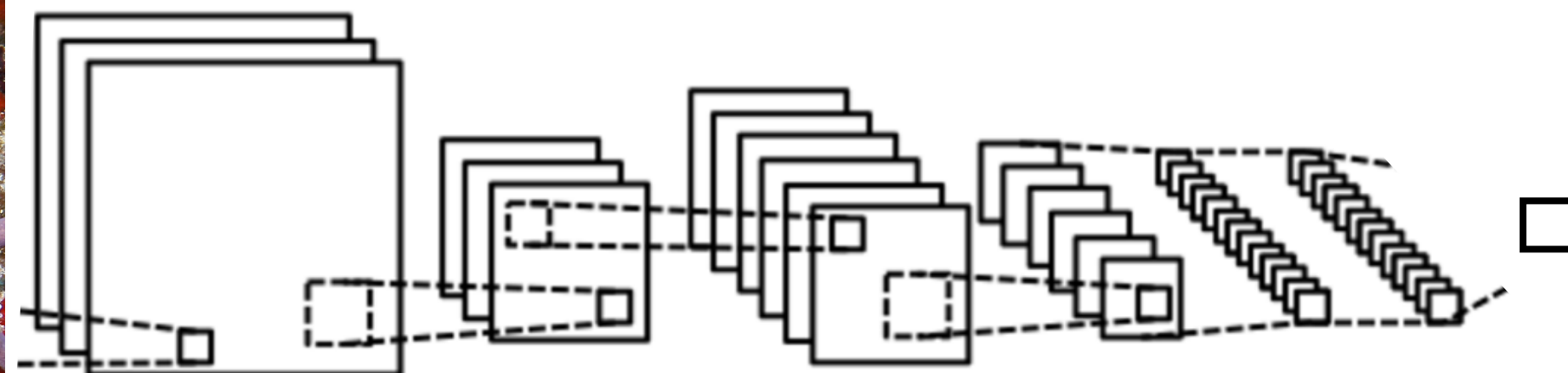
Question : quelle fonction de perte utiliser?

Idée : apprendre la fonction de perte!



2 classes :
réelle ou
générée

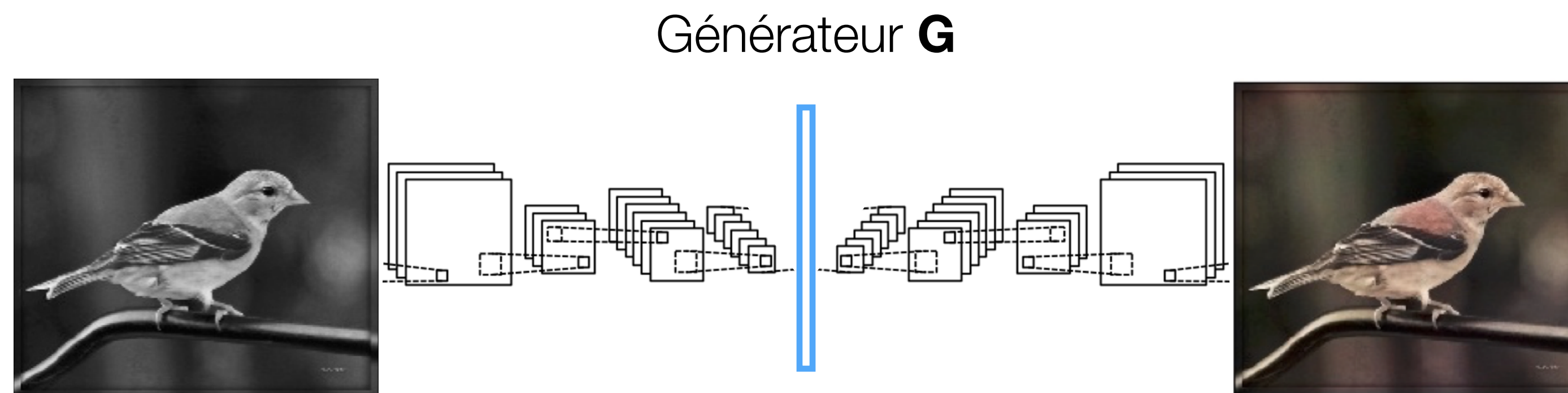
Idée : apprendre la fonction de perte!



2 classes :
réelle ou
générée

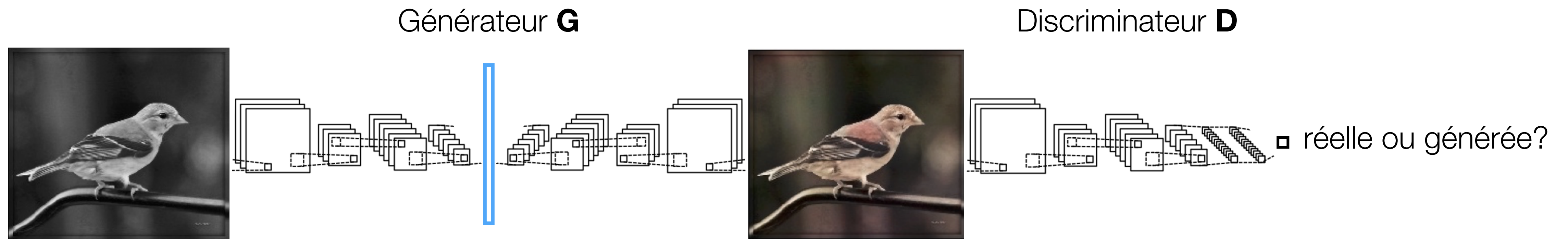
Réseau génératif antagoniste

Generative Adversarial Network (GAN)



Réseau génératif antagoniste

Generative Adversarial Network (GAN)



G essaie de générer des images pour tromper **D**

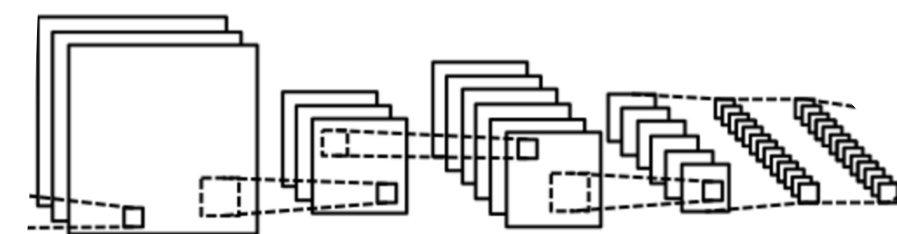
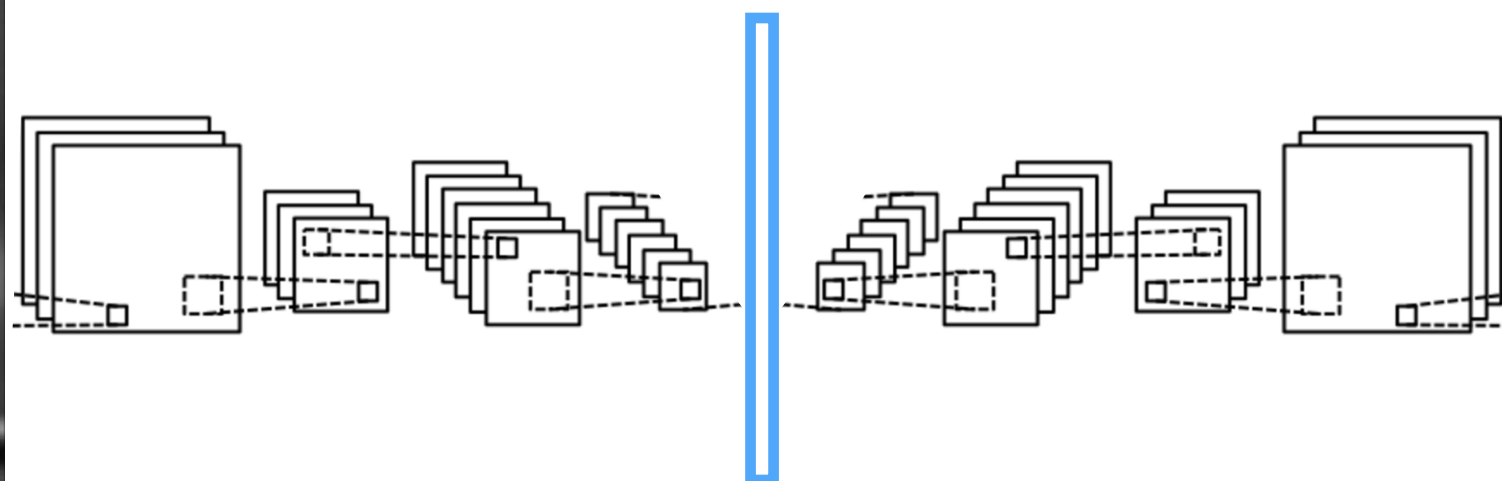
D essaie d'identifier les images générées par **G**

Images « alignées »

Si les images sont alignées (par pixel), alors on peut fournir les deux images au discriminateur!

Générateur **G**

Discriminateur **D**



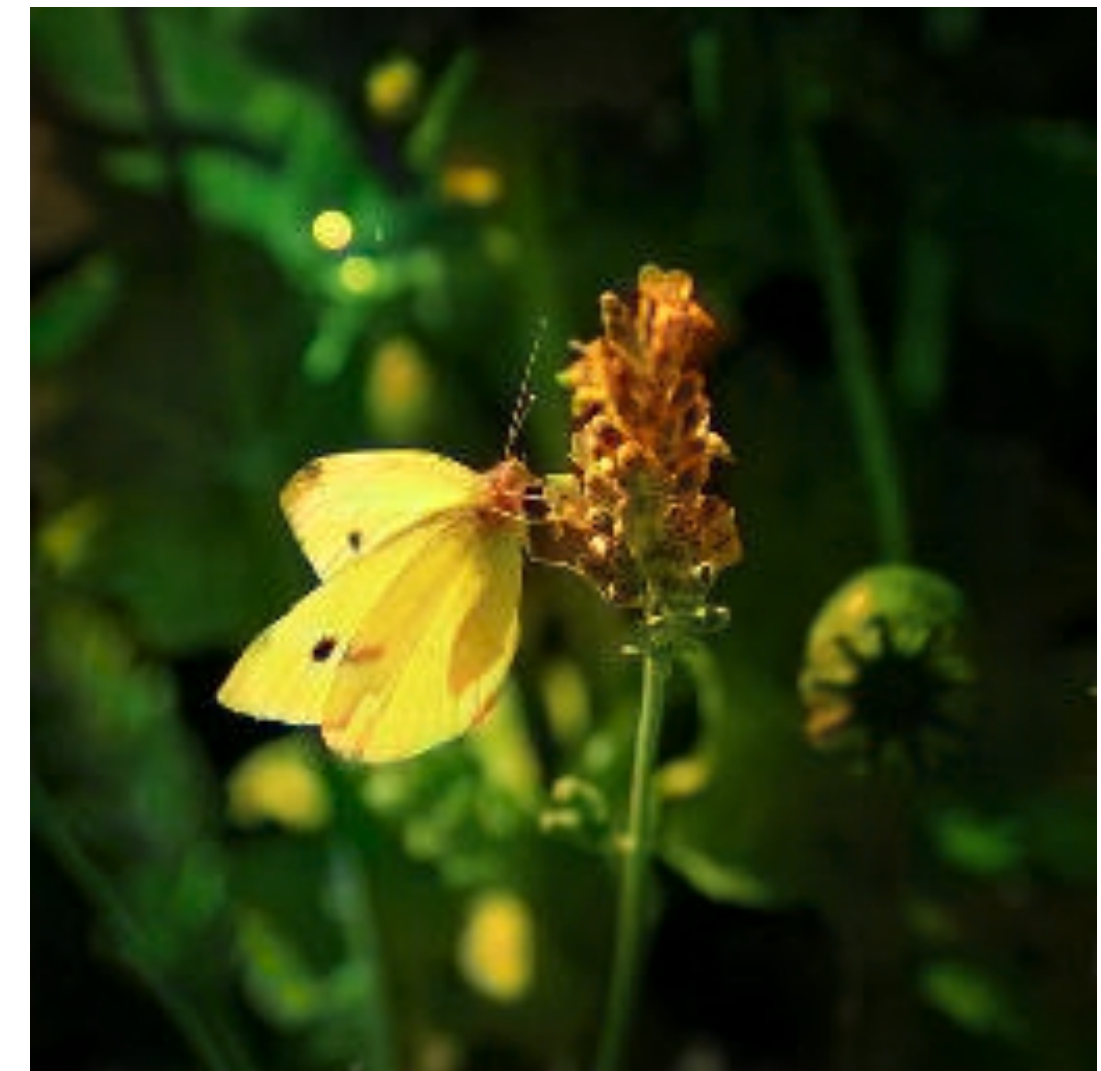
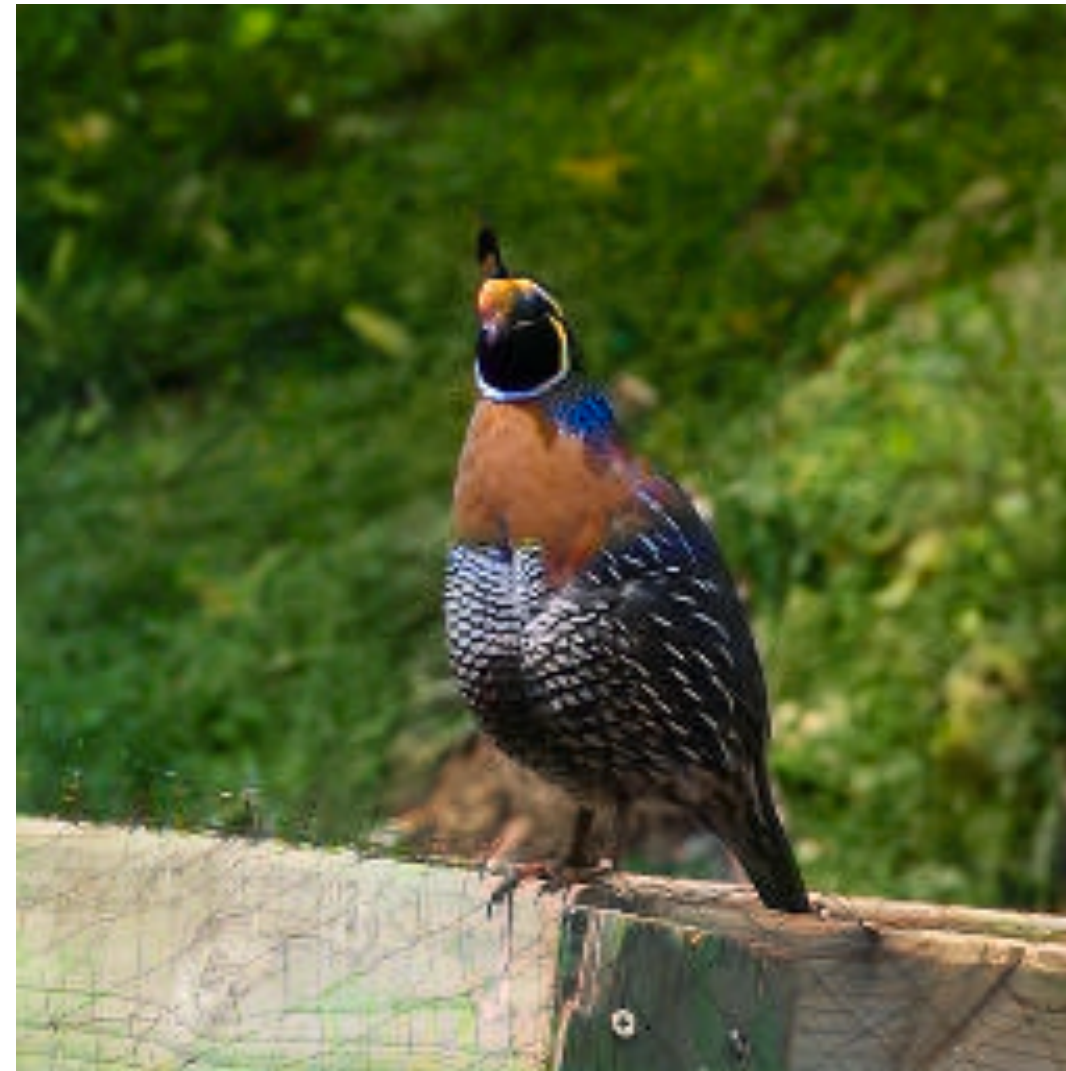
paire
réelle ou générée?

Aussi appelé « traduction d'image à image »
(*image-to-image translation*)

Images

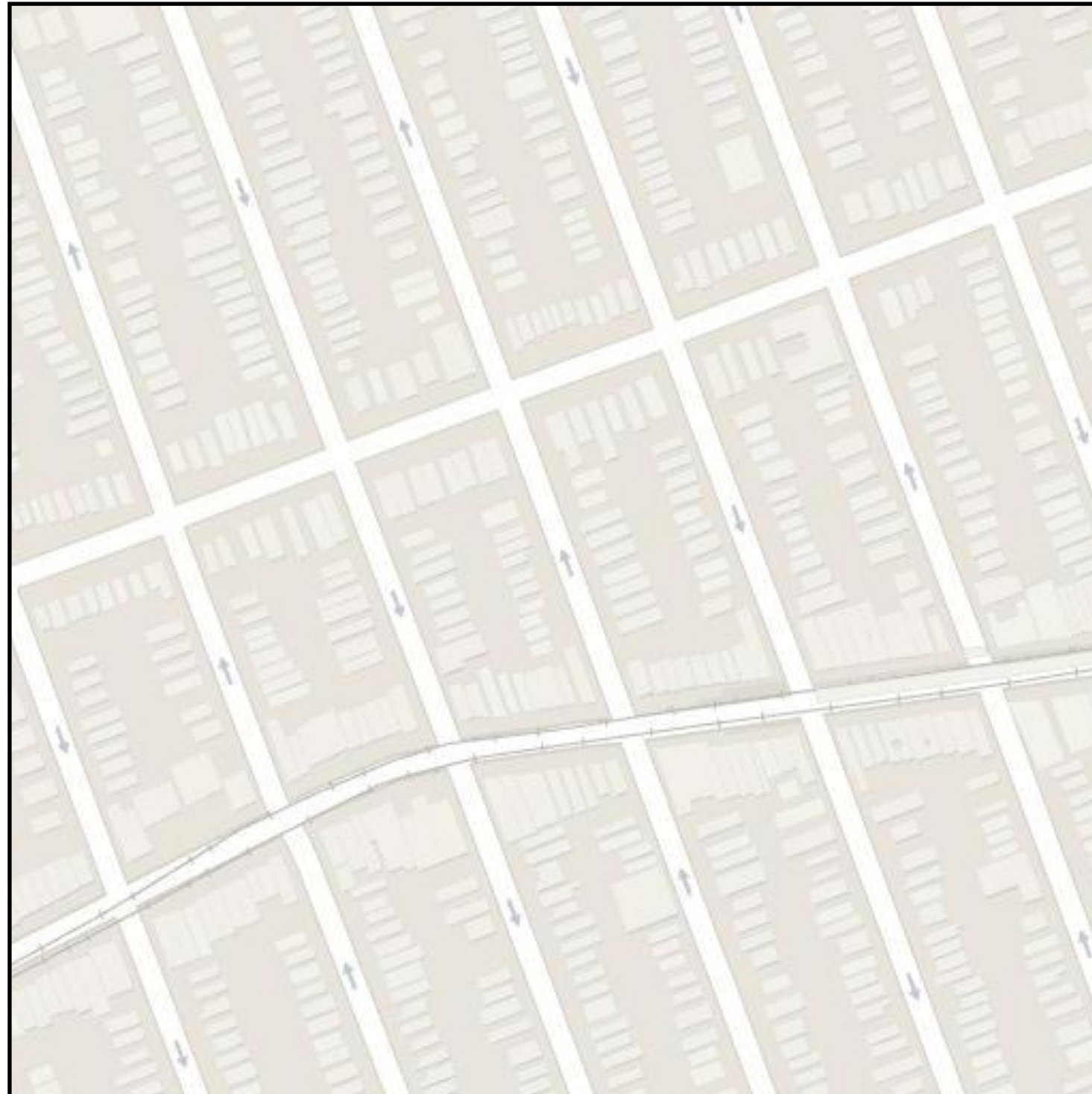


Résultats

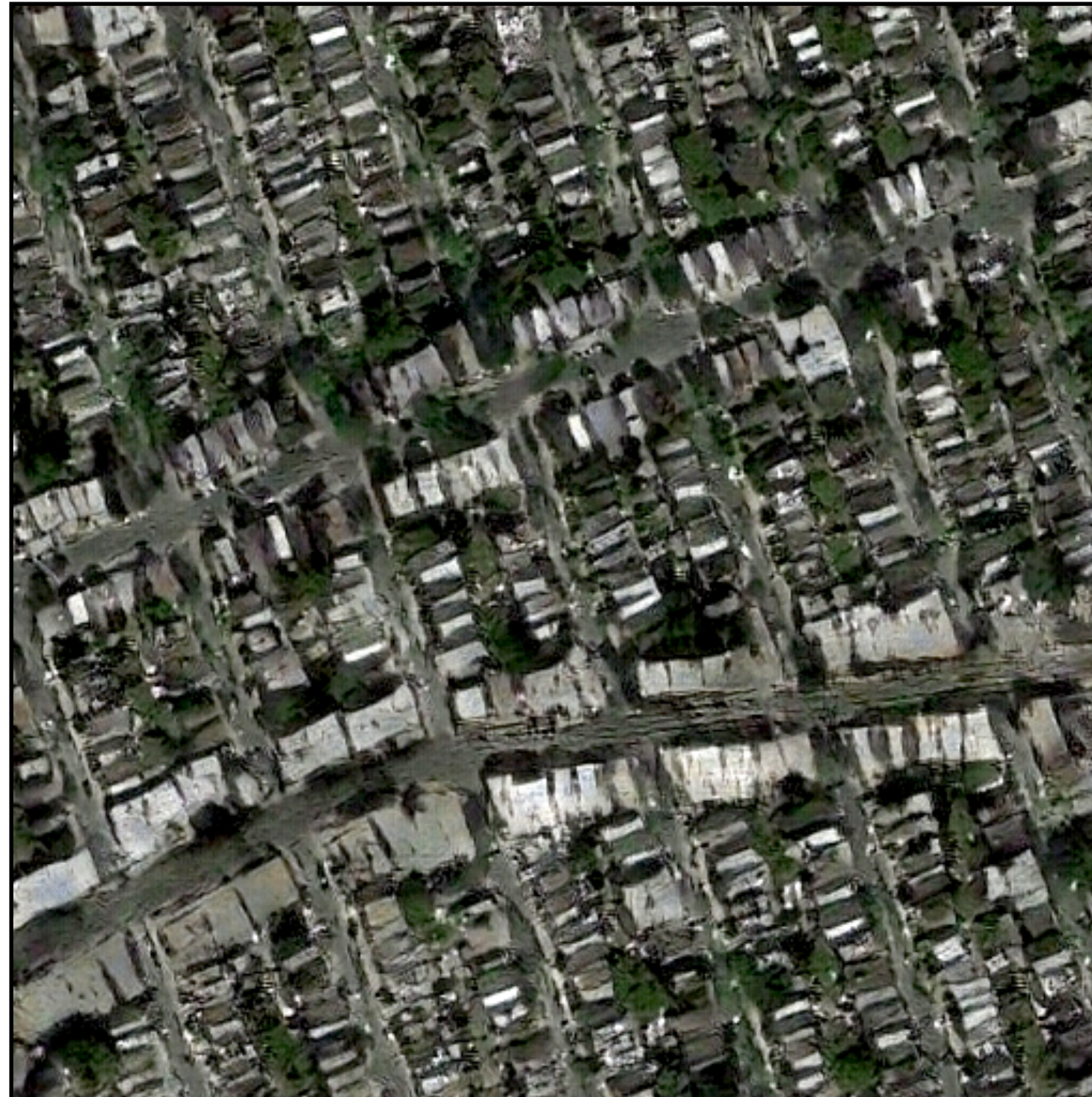


Carte vers satellite

Entrée



Sortie obtenue

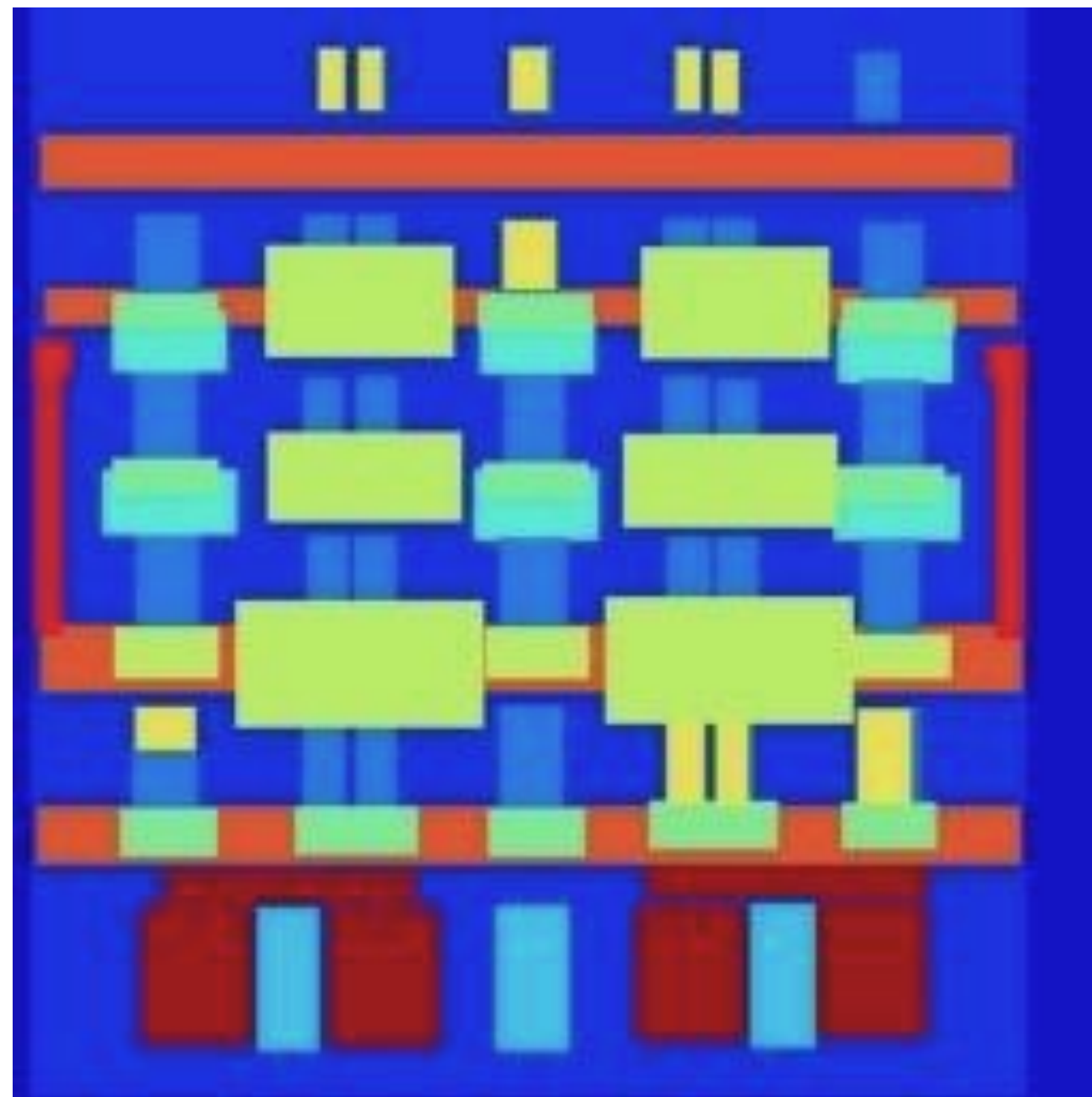


Sortie désirée



Étiquette vers façade

Entrée



Sortie obtenue



Jour vers nuit

Entrée

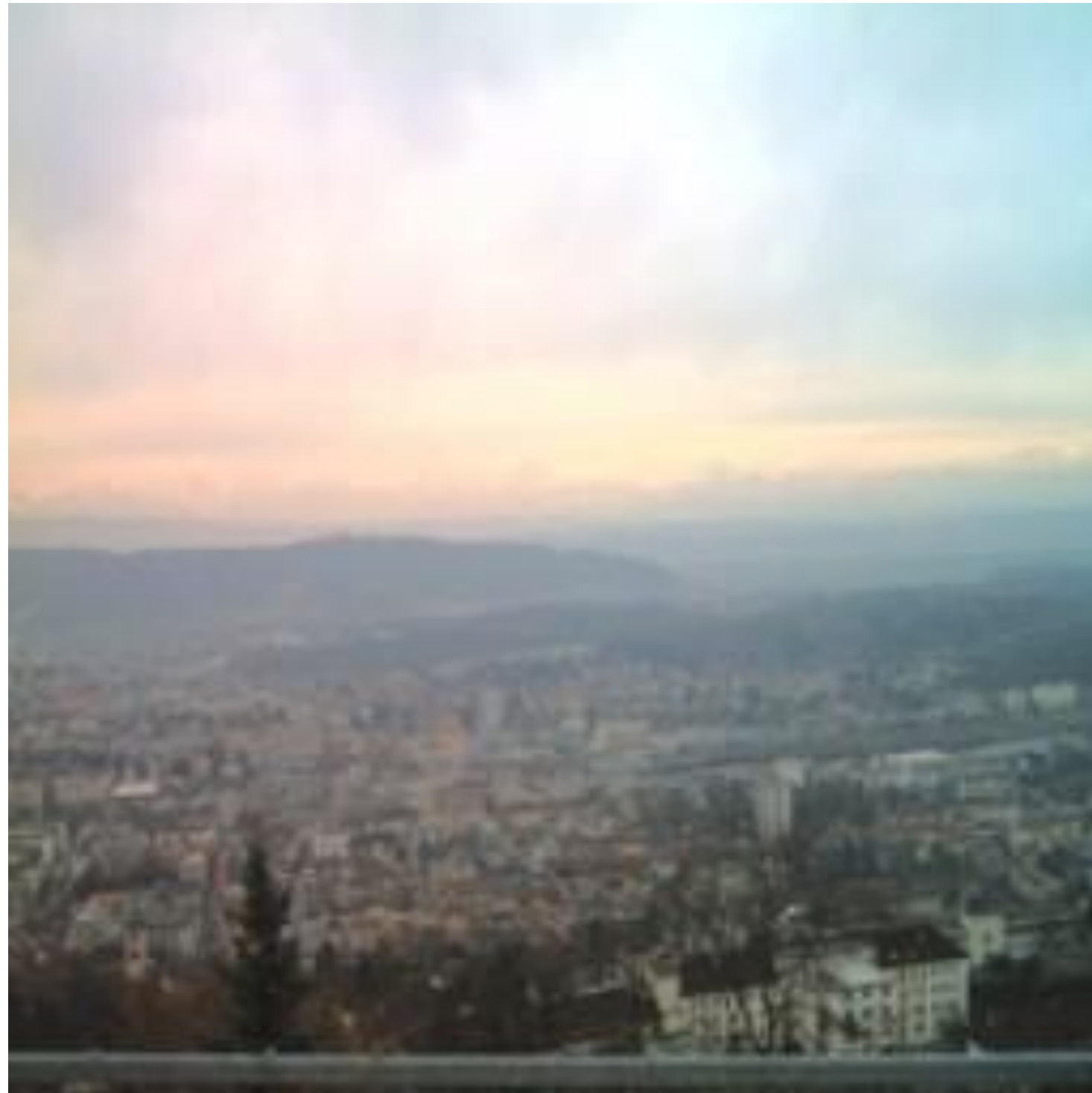


Sortie obtenue



Jour vers nuit

Entrée



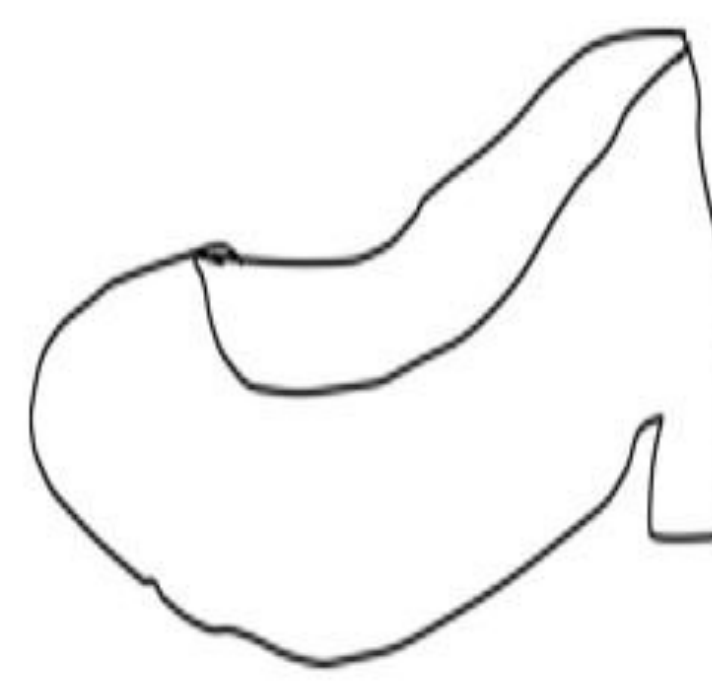
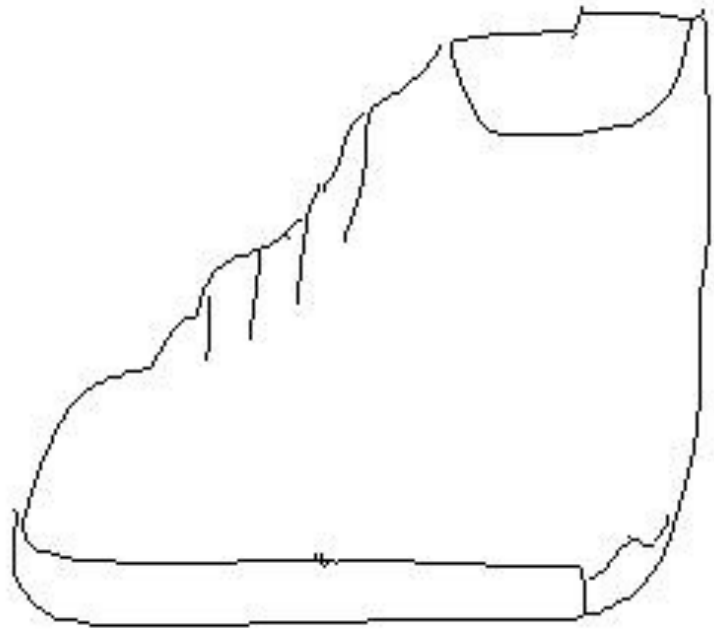
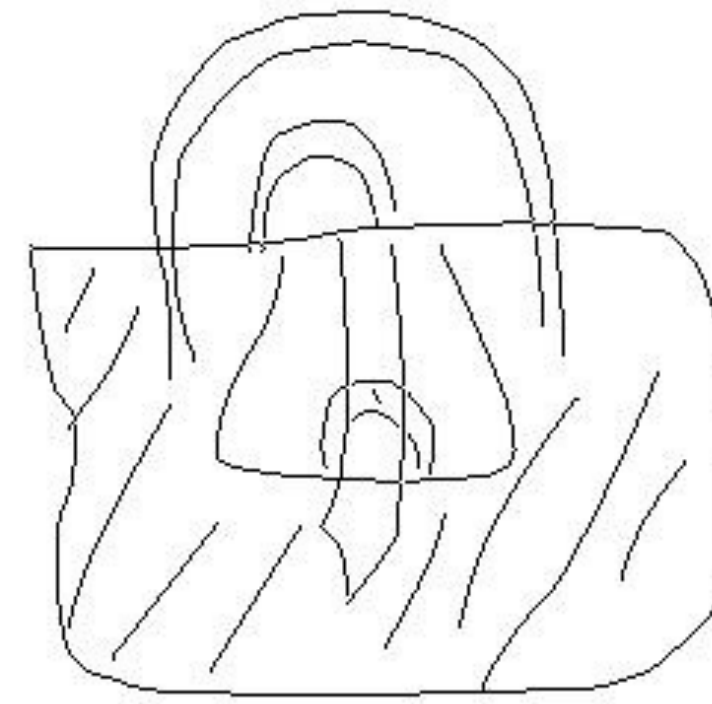
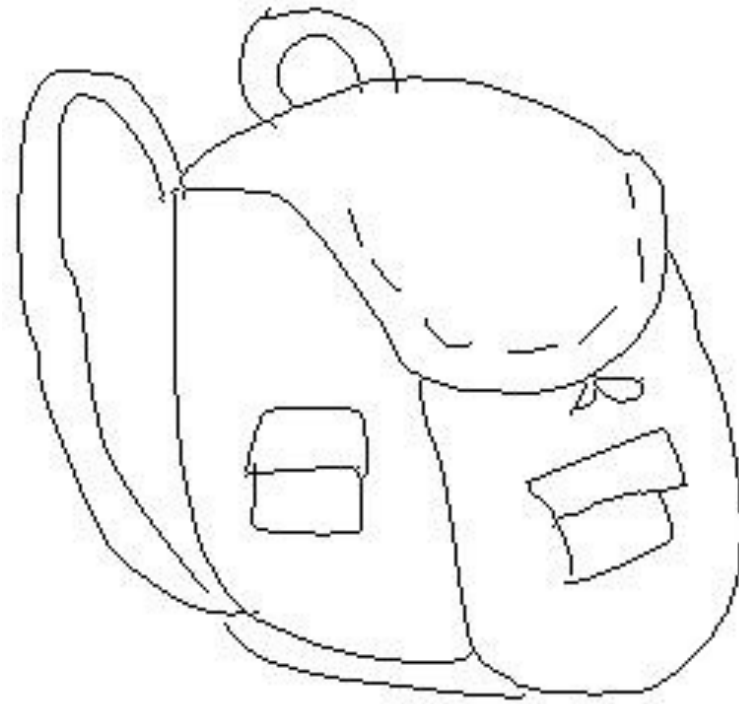
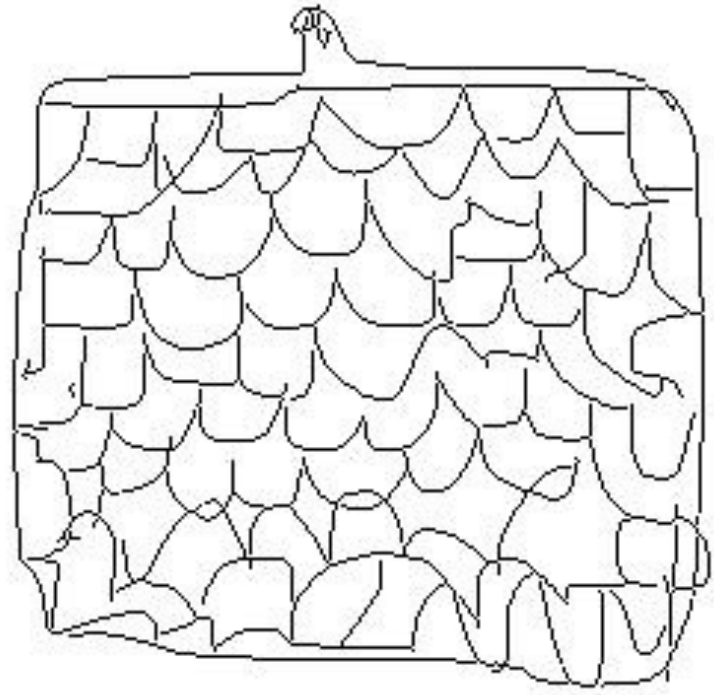
Sortie obtenue



Arêtes vers images

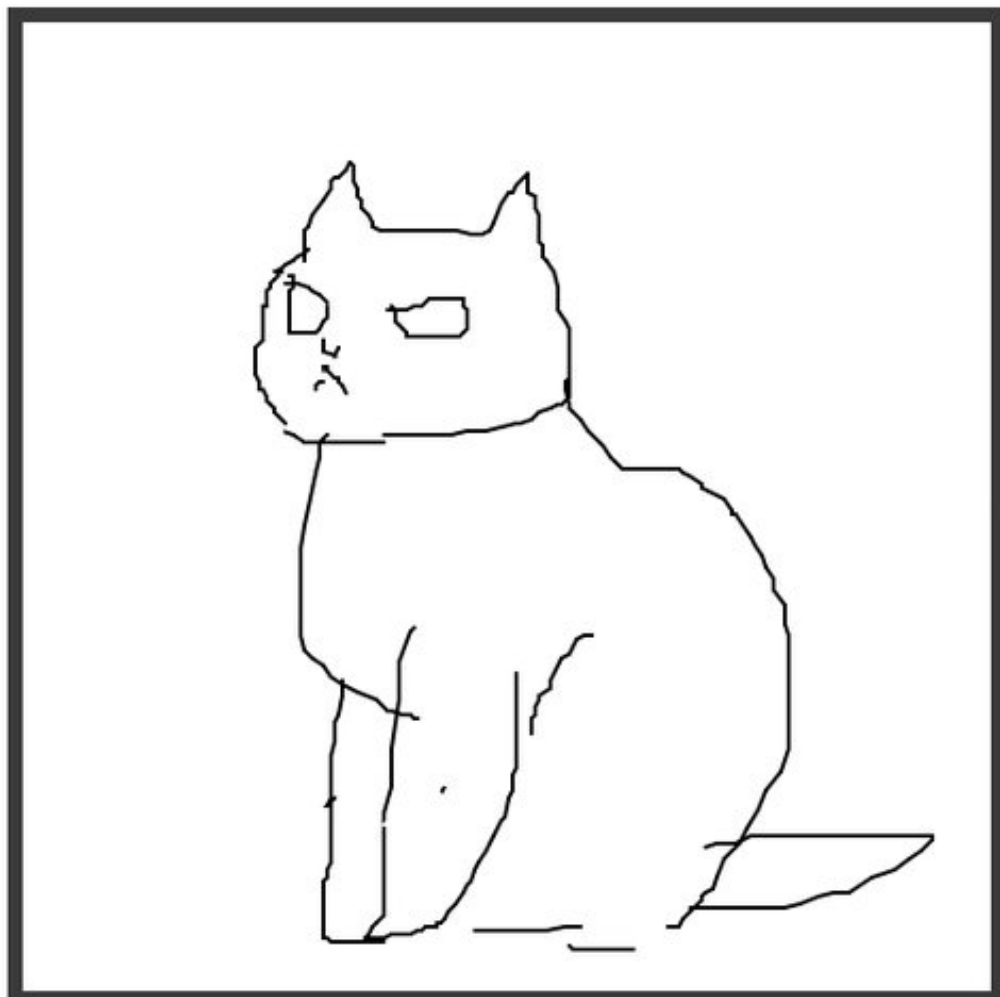


Dessins vers images



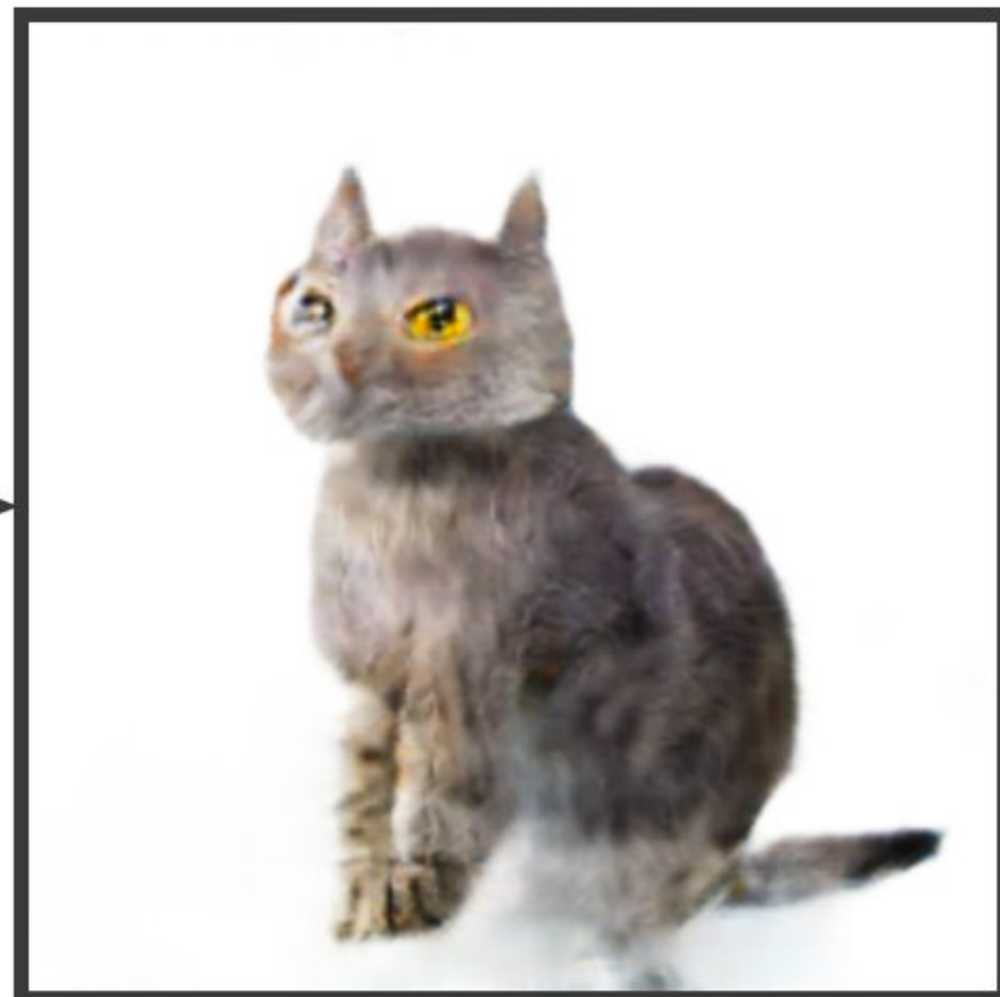
Il fallait que ça arrive... #edges2cats

INPUT



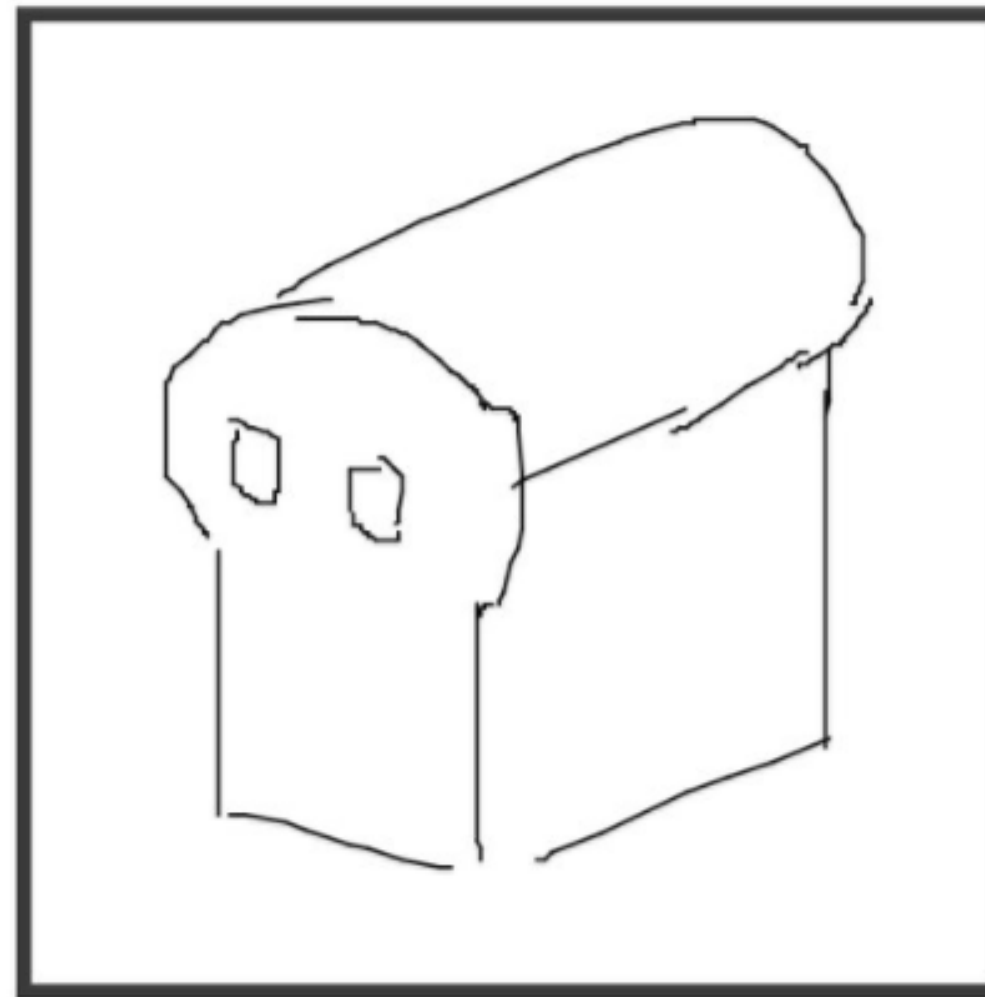
pix2pix
process

OUTPUT



@gods_tail

INPUT



pix2pix
process

OUTPUT



Ivy Tasi @ivymyt

Il fallait que ça arrive... #edges2cats



Vitaly Vidmirov @vvid

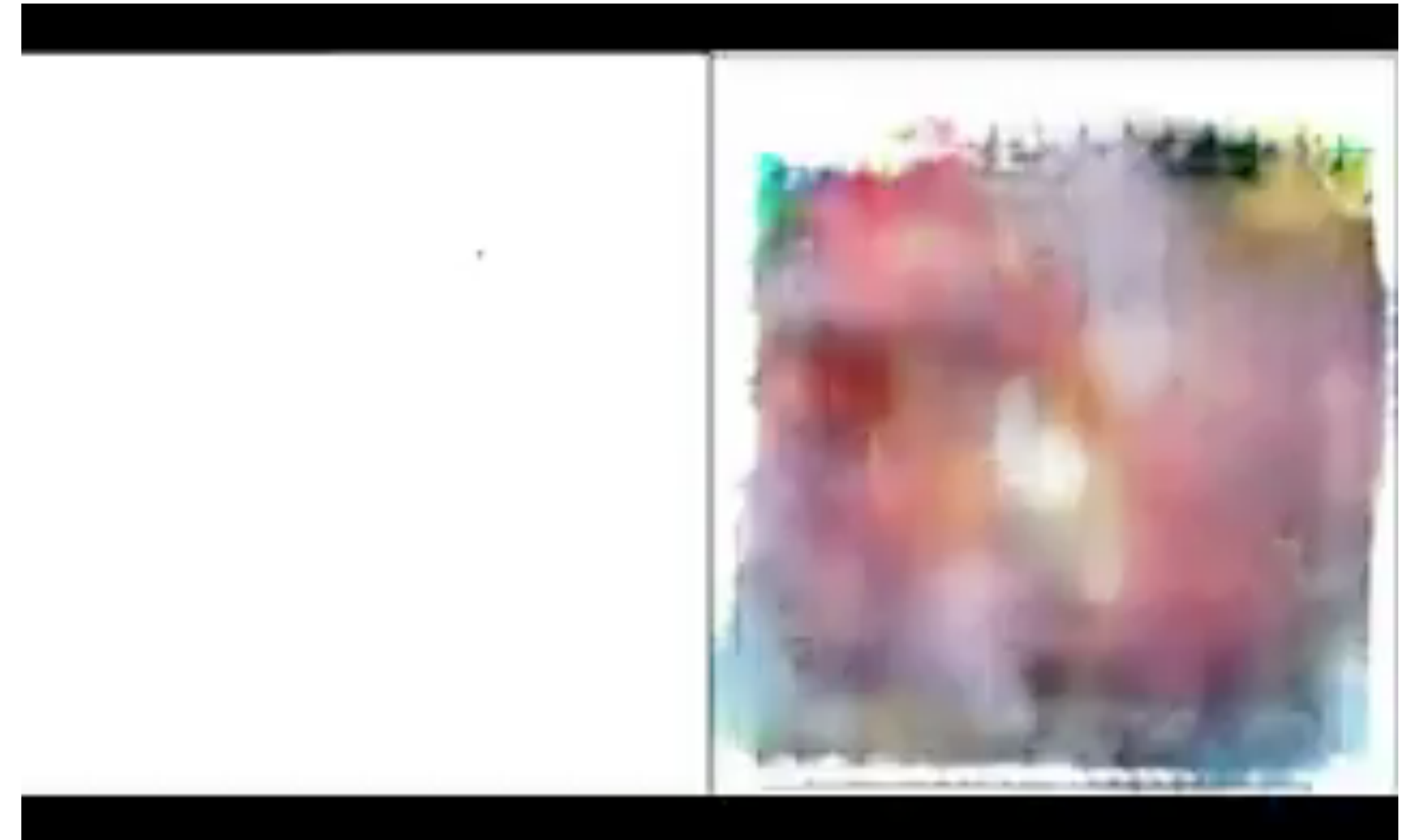


@ka92
172

Laissons libre cours à notre imagination



Mario Klingemann @quasimondo



Bertrand Gondouin @bgondouin

