

# Comment la visualisation de données peut-elle aider à la compréhension des algorithmes de machine learning ?

---

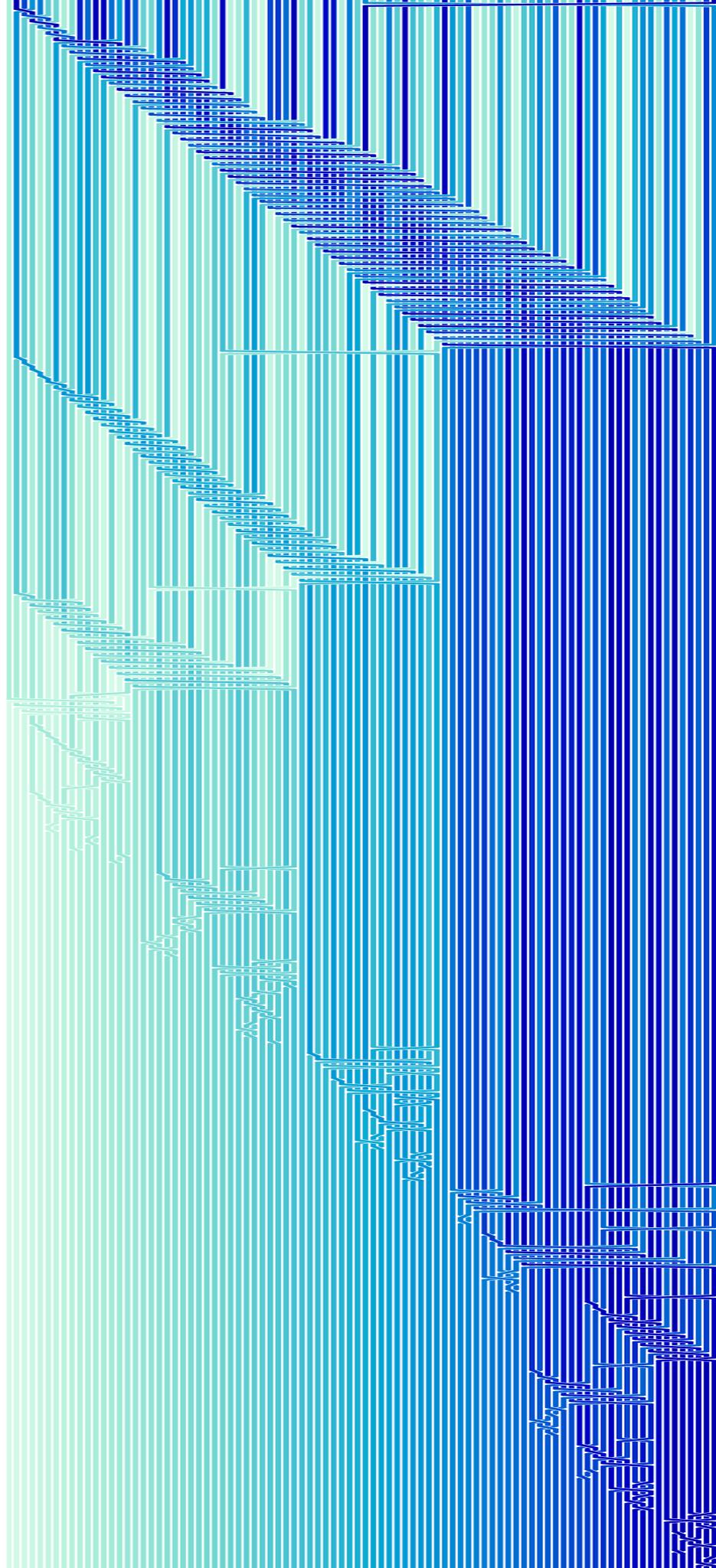
Guillaume LESAINE

01/03/2019

# Plan

---

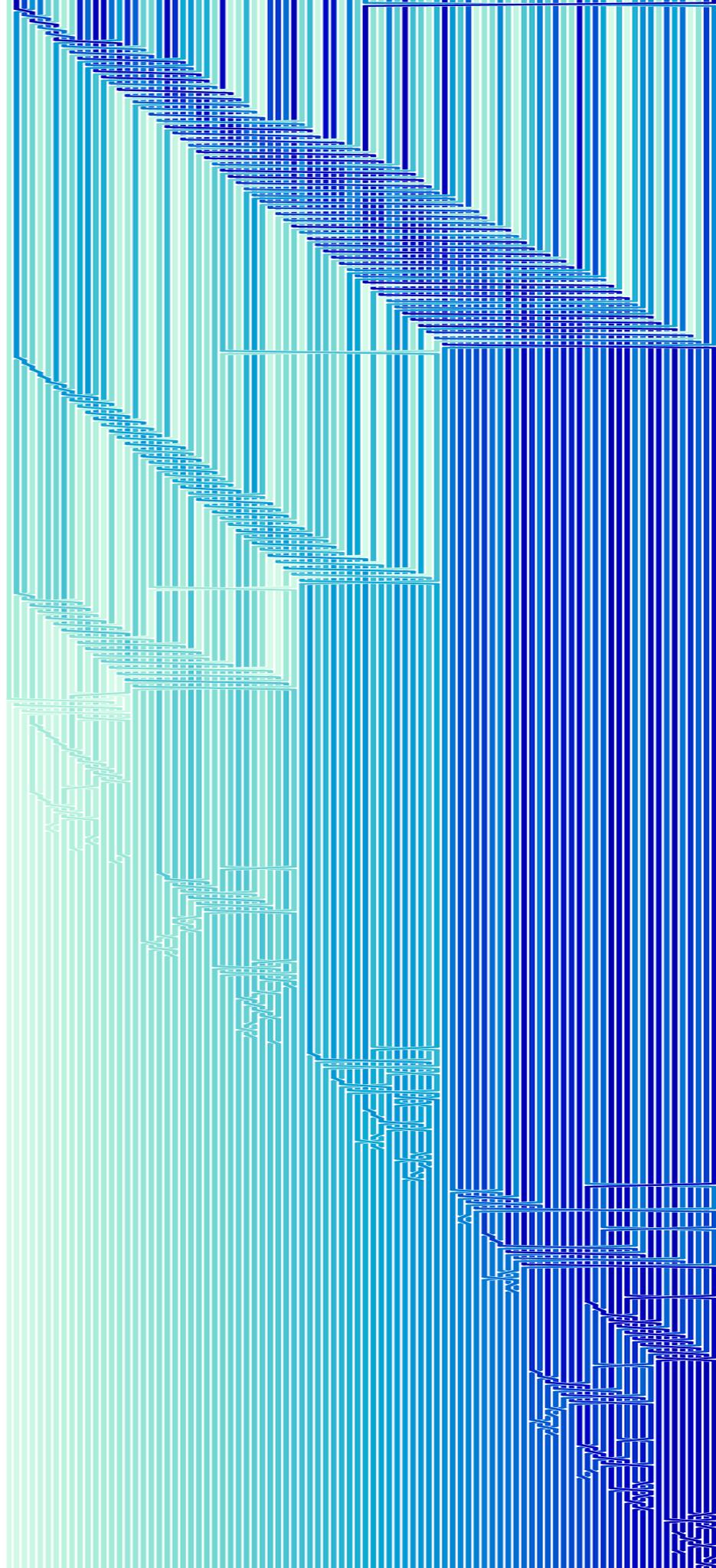
- Visualiser un algorithme
- Pourquoi visualiser l'apprentissage automatique ?
- Expliquer l'apprentissage automatique
- Limites



# Plan

---

- Visualiser un algorithme
  - Difficulté
  - Boîte noire
  - Exemple
- Pourquoi visualiser l'apprentissage automatique ?
- Expliquer l'apprentissage automatique
- Limites



# Difficulté de visualiser un algorithme

---

- Visualisation

“L'utilisation de représentations visuelles de données pour amplifier la cognition.”[1]

# Difficulté de visualiser un algorithme

- Visualisation

“L'utilisation de représentations visuelles de données pour amplifier la cognition.”[1]



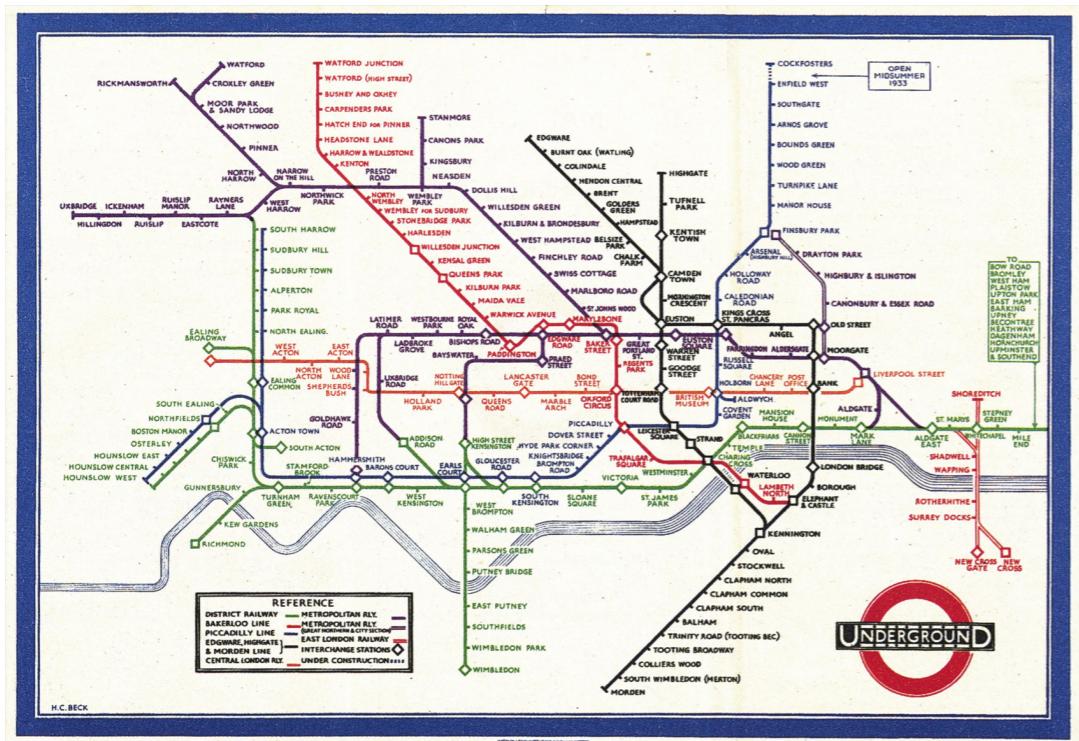
# Difficulté de visualiser un algorithme

- Visualisation

“L'utilisation de représentations visuelles de données pour amplifier la cognition.”[1]

- Algorithme :

“Ensemble de règles opératoires dont l'application permet de résoudre un problème énoncé au moyen d'une somme finie d'opérations.”[2]



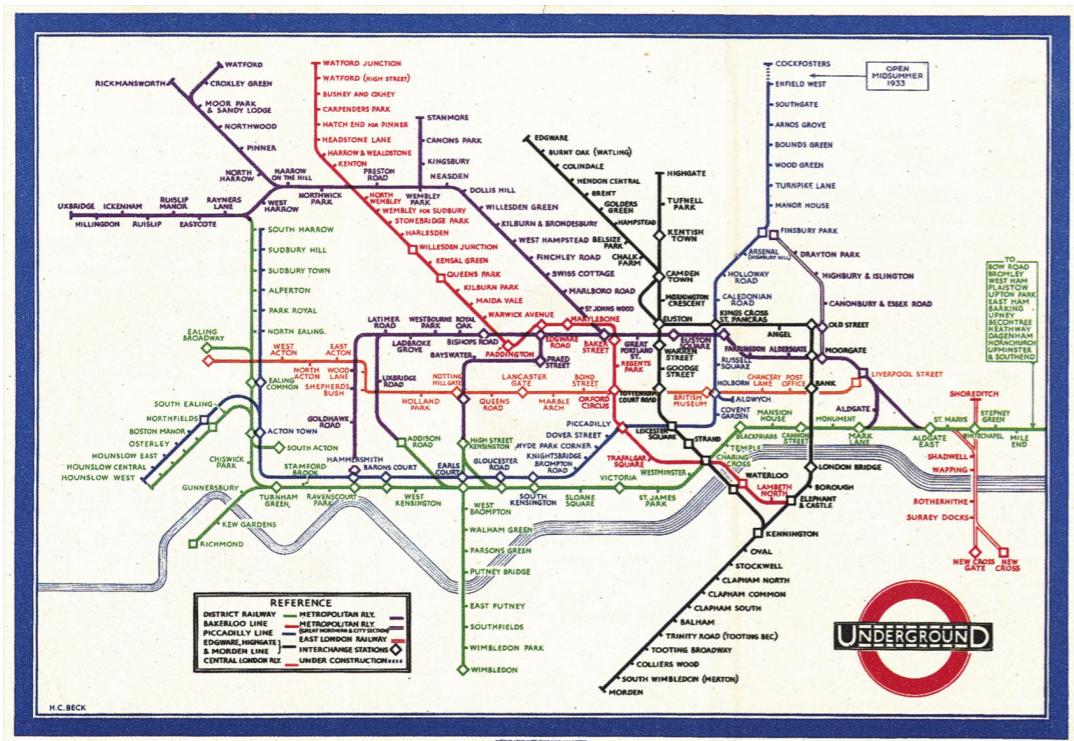
# Difficulté de visualiser un algorithme

- Visualisation

“L'utilisation de représentations visuelles de données pour amplifier la cognition.”[1]

- Algorithme :

“Ensemble de règles opératoires dont l'application permet de résoudre un problème énoncé au moyen d'une somme finie d'opérations.”[2]



#### Proposed Algorithm:

**Input :**  $S = \{S^1, S^2, \dots, S^N\}$ , variables  $S^i = \{S_1^i, S_2^i, \dots, S_n^i\}$ .

**Output :** a set of  $i$  Lift Ratio $i$  values for each MVTS

**For each multivariate time series**

Discretized data using Equal Frequency for  $S^i$

For each variables ( $v_1$ ) in MVTS

get intra-patterns Lift ( $s_{j,t-\tau}^i \Rightarrow s_{j,t}^i$ ) ( $1 \leq \tau < t$ )

For each variables  $v_2$  besides  $v_1$  in MVTS

get inter-patterns Lift ( $s_{j,t-\tau}^i \Rightarrow s_{j,t}^i$ ) ( $1 \leq \tau < t$ )

Result += inter-patterns

(Result is one set of Lift Ratio for one MVTS)

**End**

Result += inter-patterns

**End**

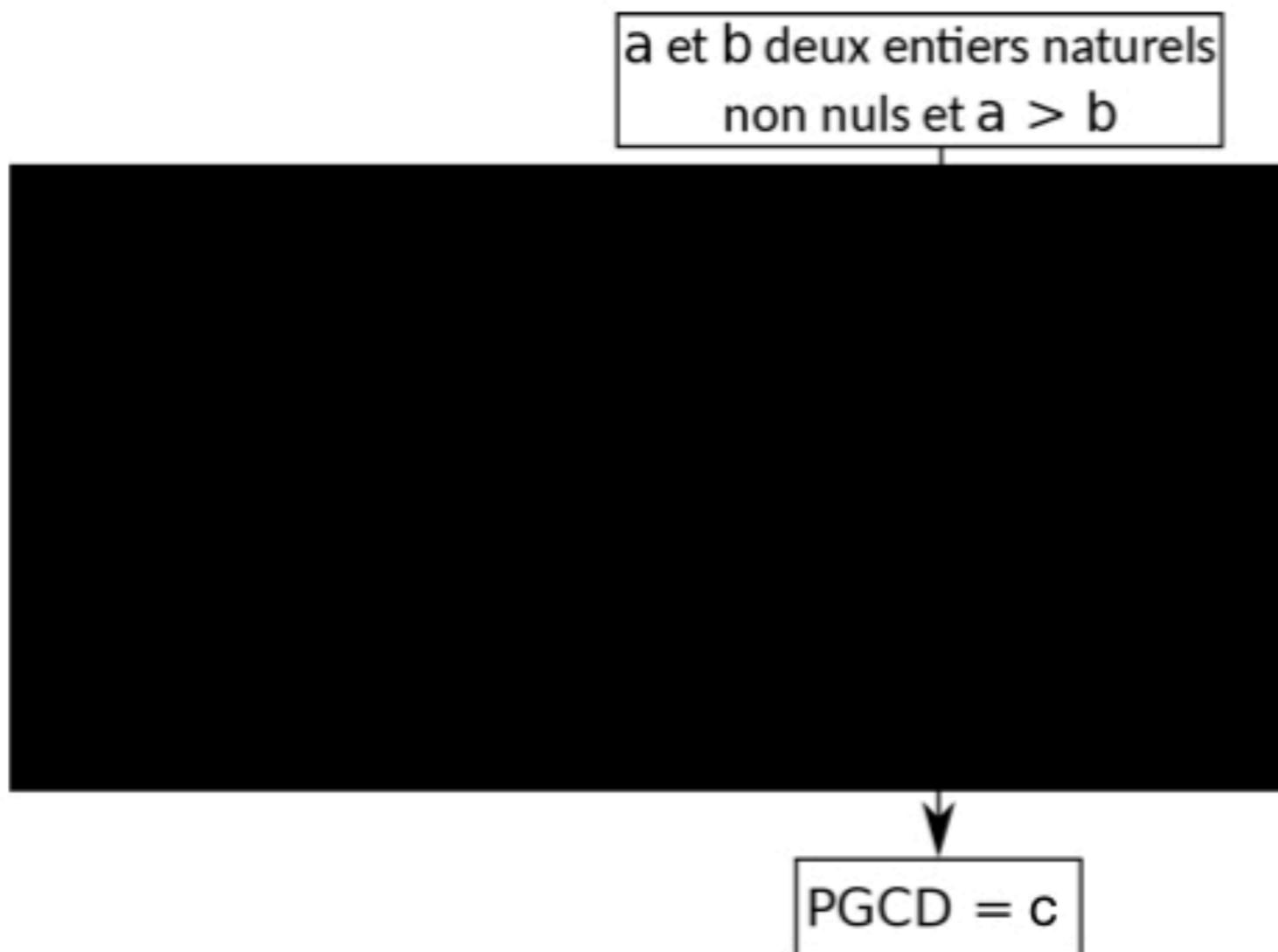
finalResult += Result

**End** (finalResult are all Lift Ratio for all MVTSs)

**K-means clustering continue.**

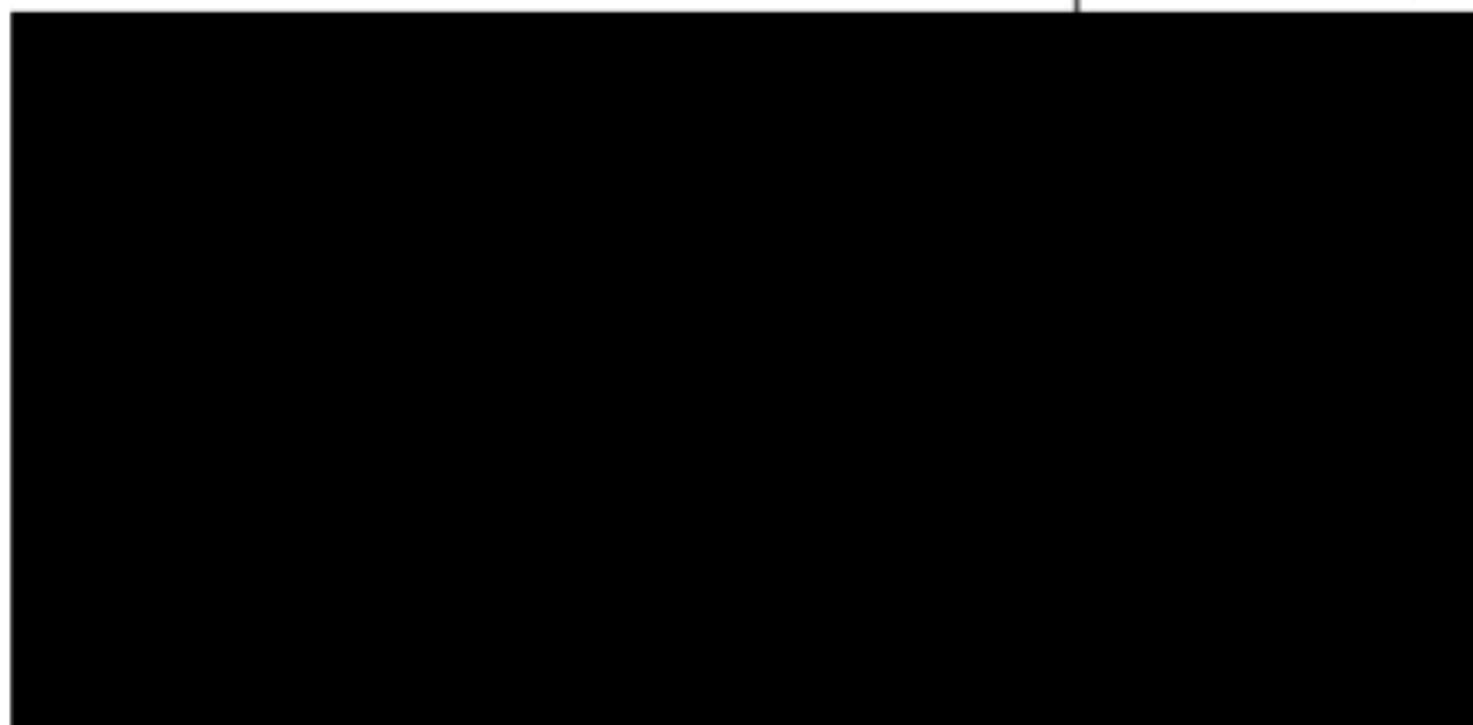
# Boîte noire

---



# Boîte noire

---



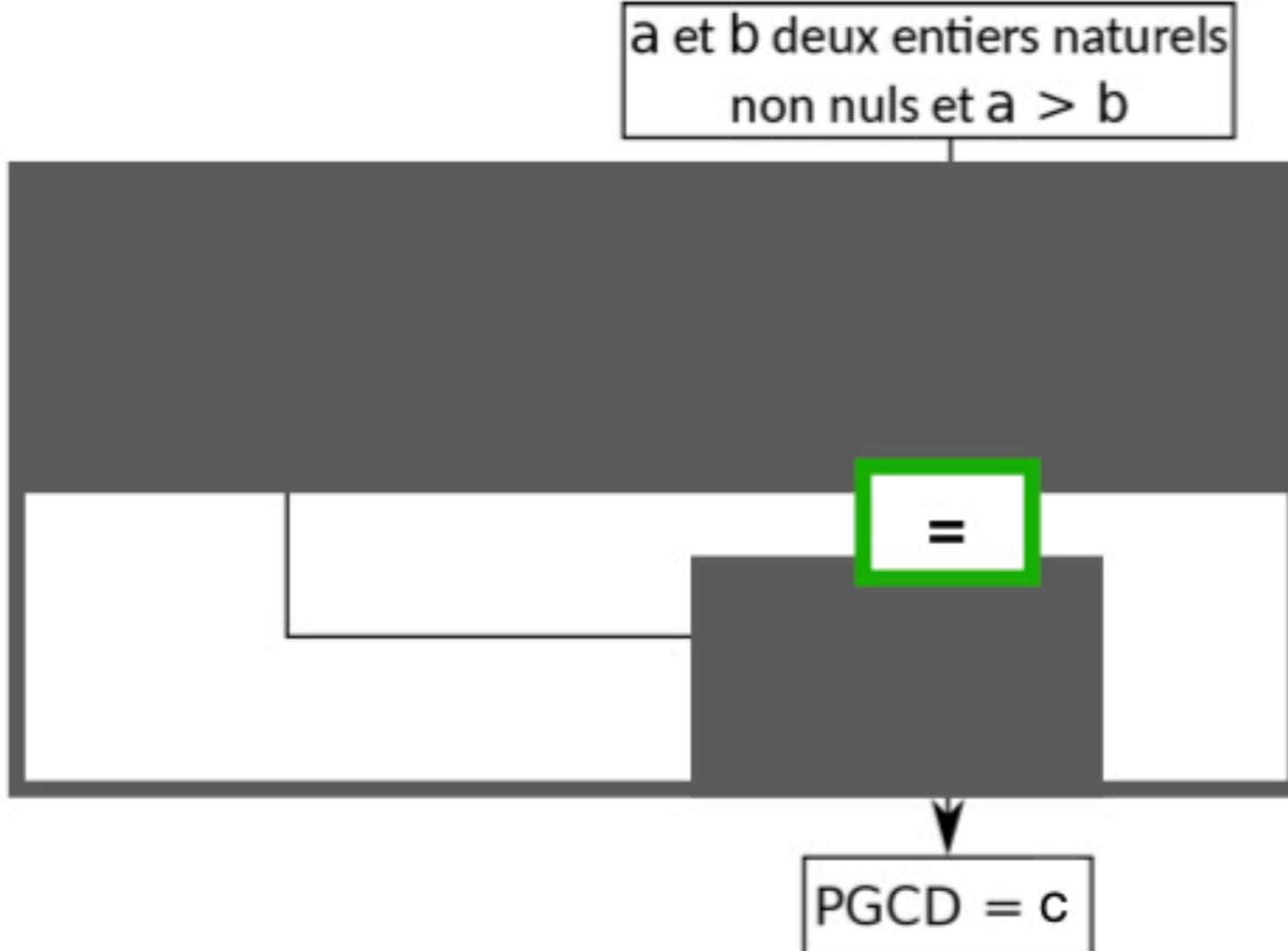
PGCD = c

>>> PGCD(72,26)

>>> 2

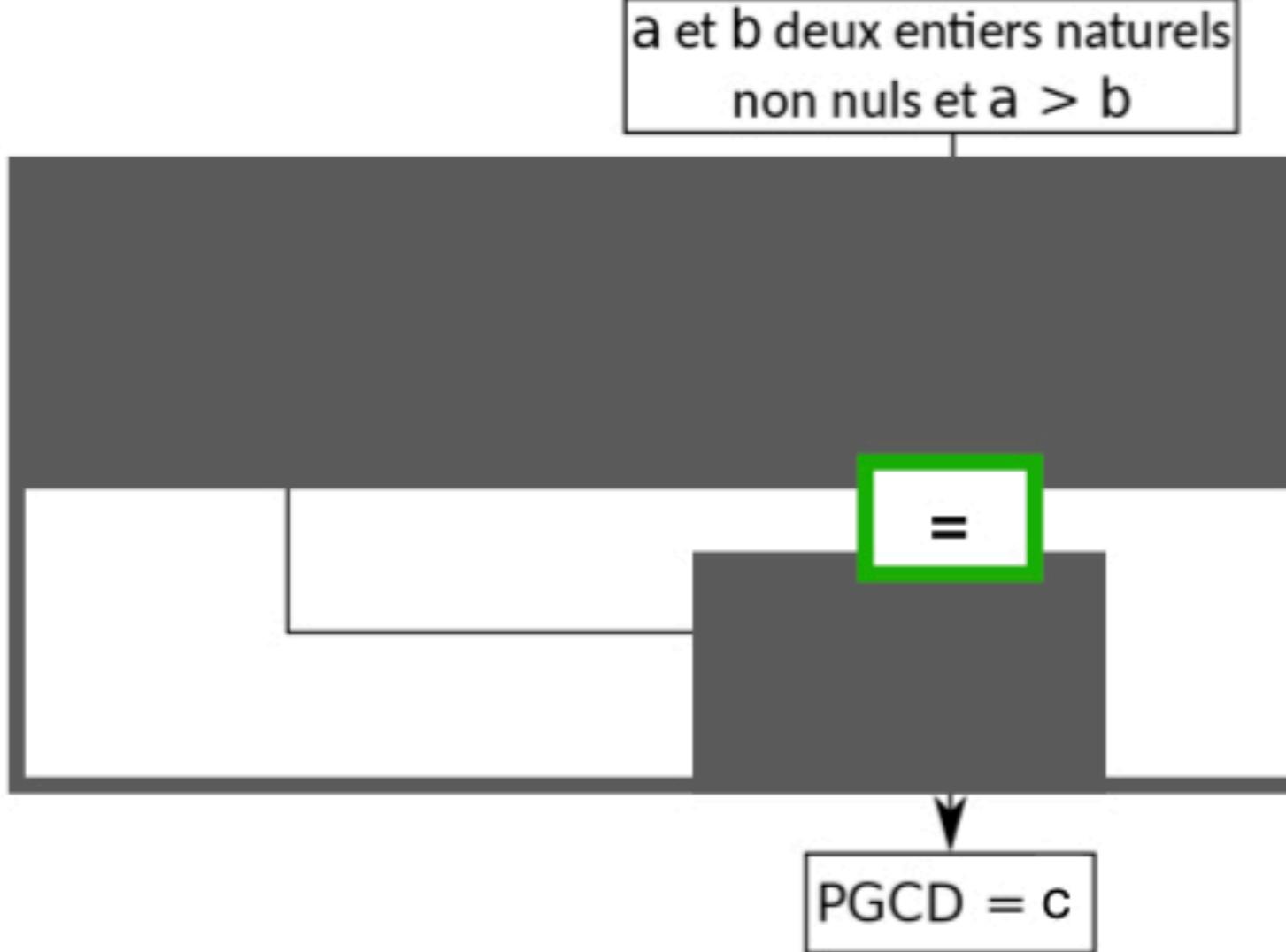
# Boîte grise

---



# Boîte grise

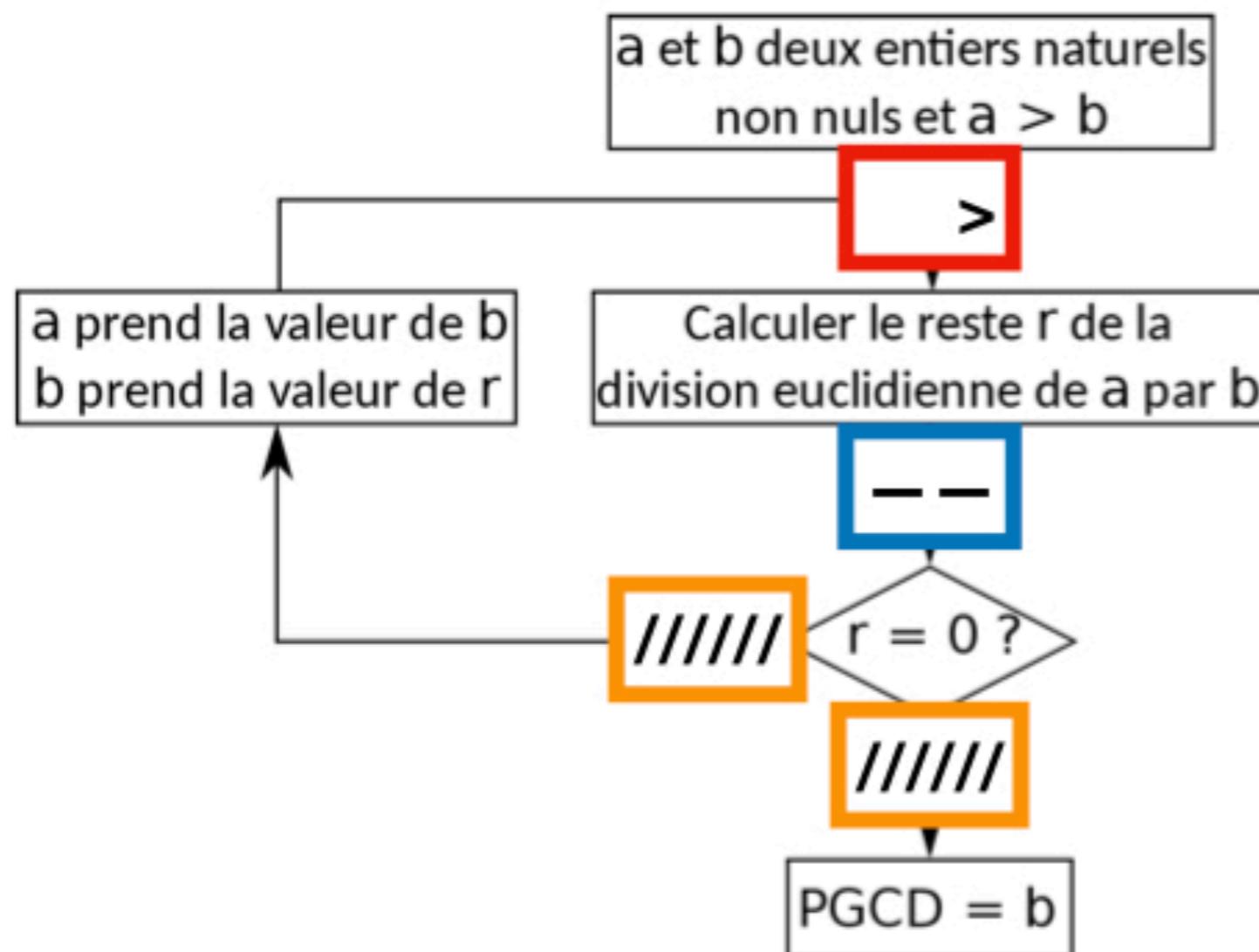
---



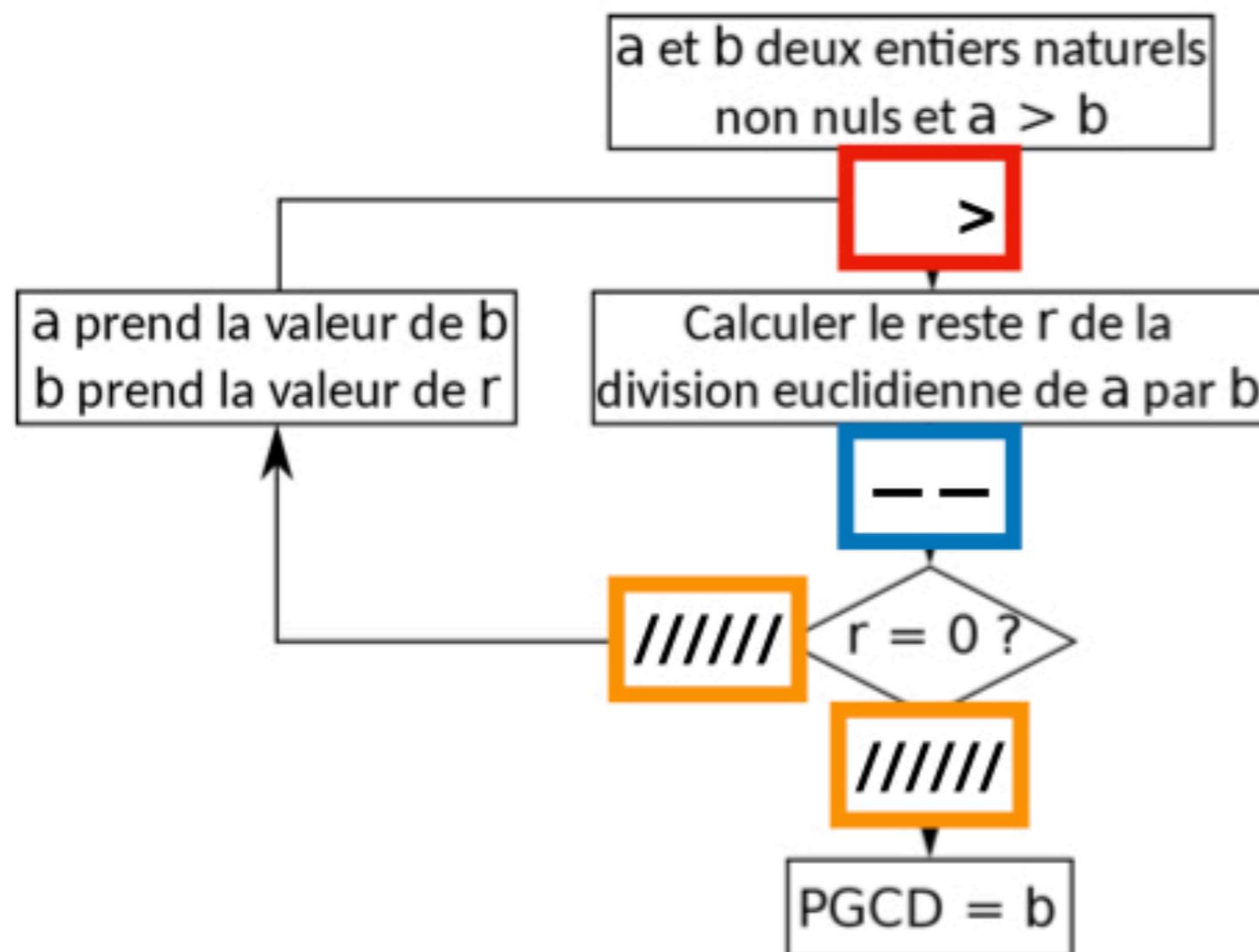
```
>>> PGCD(72,26)
= 26
= 20
= 6
= 2
>>> 2
```

# Boîte blanche

---



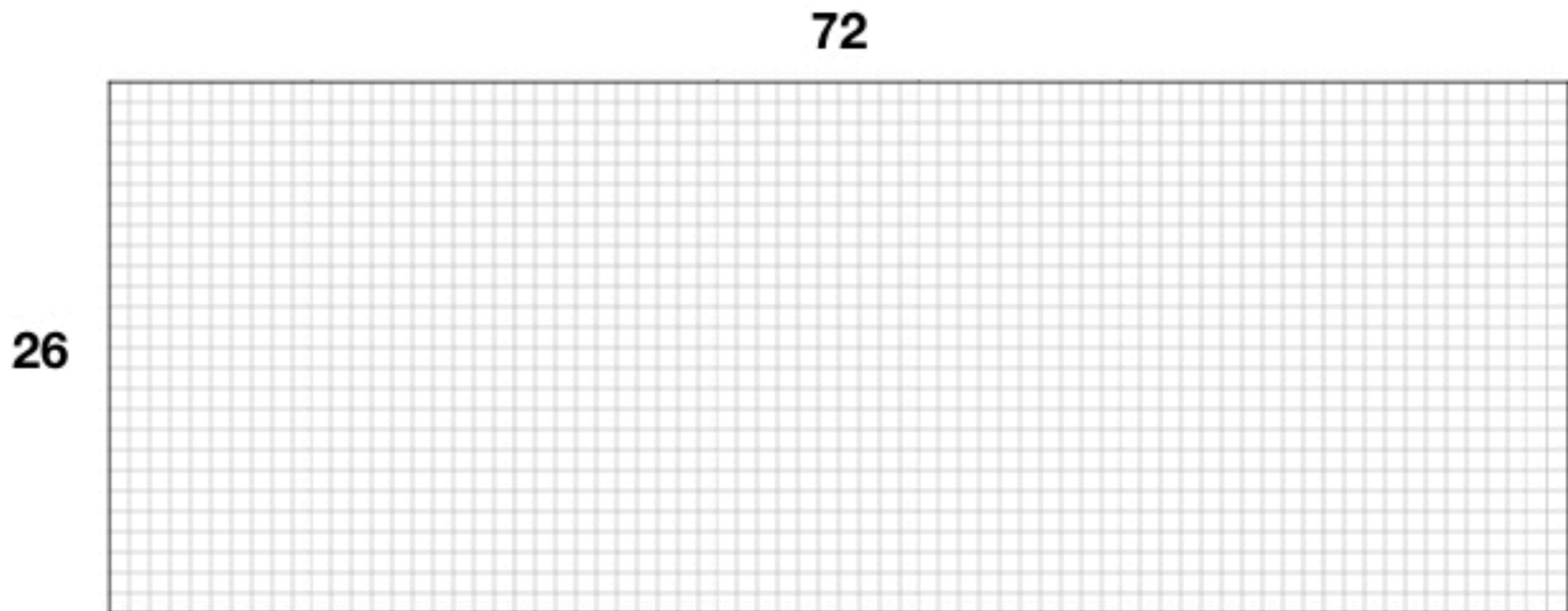
# Boîte blanche



```
>>> PGCD(72,26)
> a = 72  b = 26
--- r = 20
||||| NON
> a = 26  b = 20
--- r = 6
||||| NON
> a = 20  b = 6
--- r = 2
||||| NON
> a = 6   b = 2
--- r = 0
||||| OUI
>>> 2
```

# Exemple : Algorithme d'Euclide du PGCD

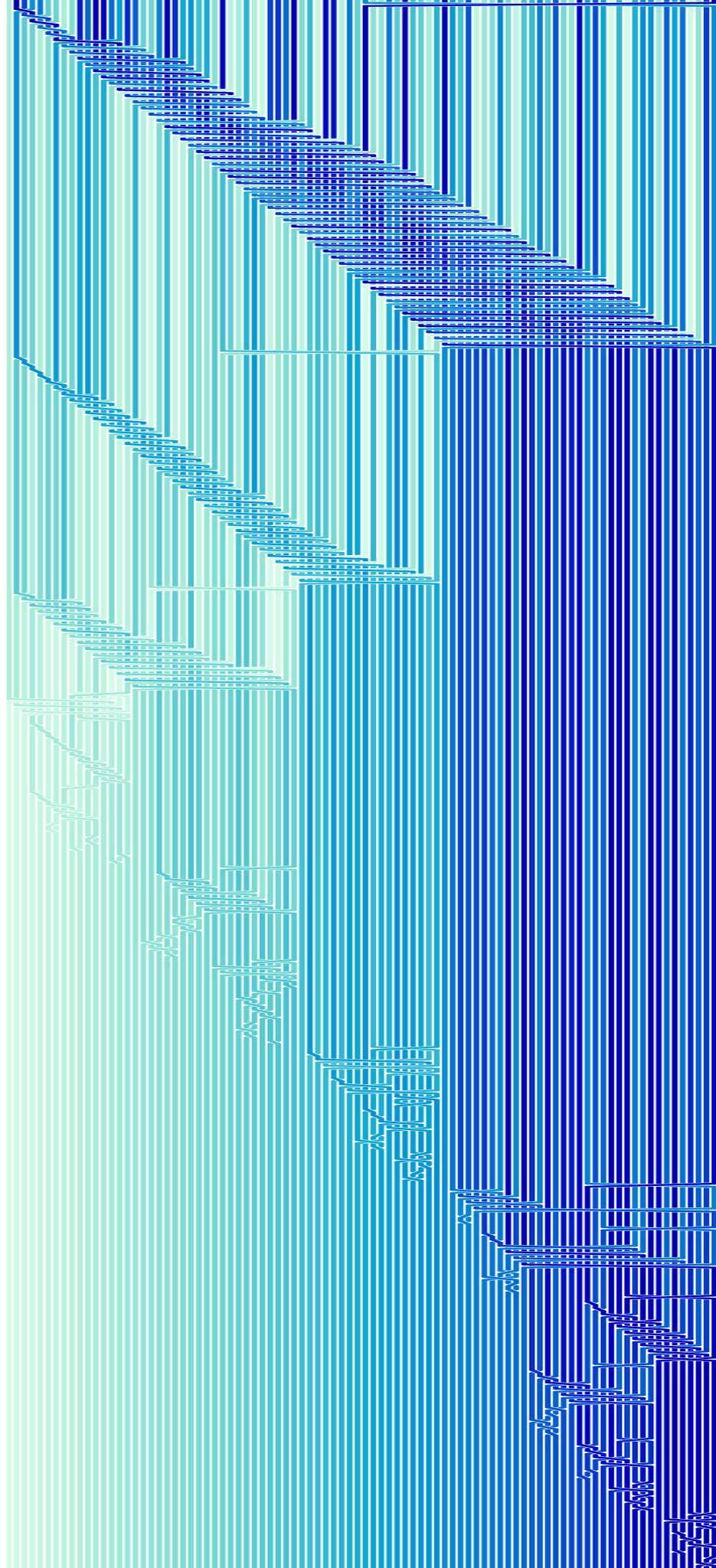
---



# Plan

---

- Visualiser un algorithme
- Pourquoi visualiser l'apprentissage automatique ?
  - Un problème
  - Une solution ?
  - Des raisons
- Expliquer l'apprentissage automatique
- Limites



# Apprentissage automatique

---

Apprentissage automatique :

*“In machine learning models, it’s not a programmer that writes the program, it’s a machine that writes the program and how does the machine know to write the program? Through data”[3]*



Peter Norvig - Head of Research at Google

# Un problème

2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019



YouTube



f



in



tw



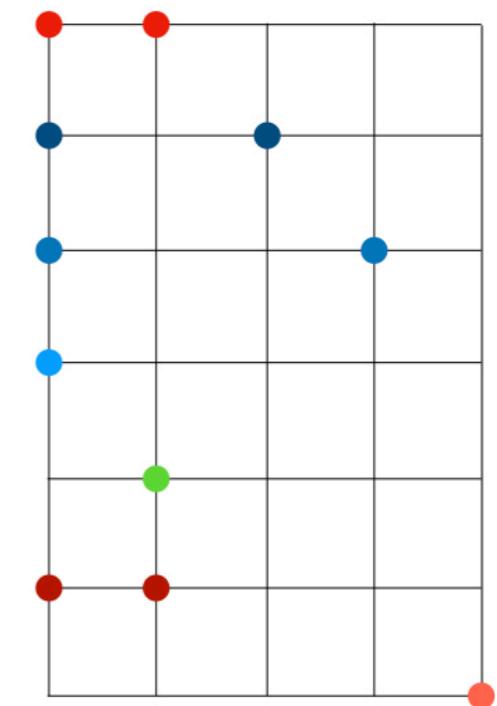
Spotify



NETFLIX



tinder



# Un problème

2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019



YouTube



f



in



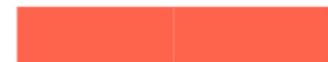
tw



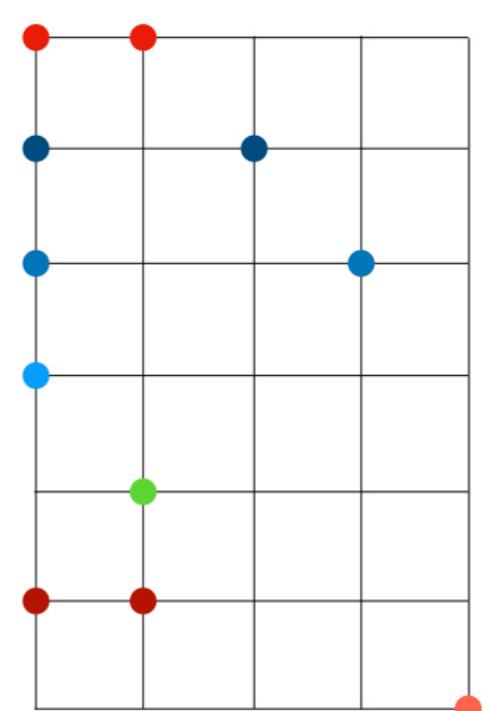
Spotify



NETFLIX



tinder



RENAULT



SÉCURITÉ SOCIALE  
l'Assurance  
Maladie



# Une solution ? RGPD : Droit à l'explication

---

Article 13 et 14 - Dans le cas d'une prise de décision automatisée :

*“des informations utiles concernant la logique sous-jacente, ainsi que l’importance et les conséquences prévues de ce traitement pour la personne concernée.”[4]*



# Raisons de visualiser l'apprentissage automatique

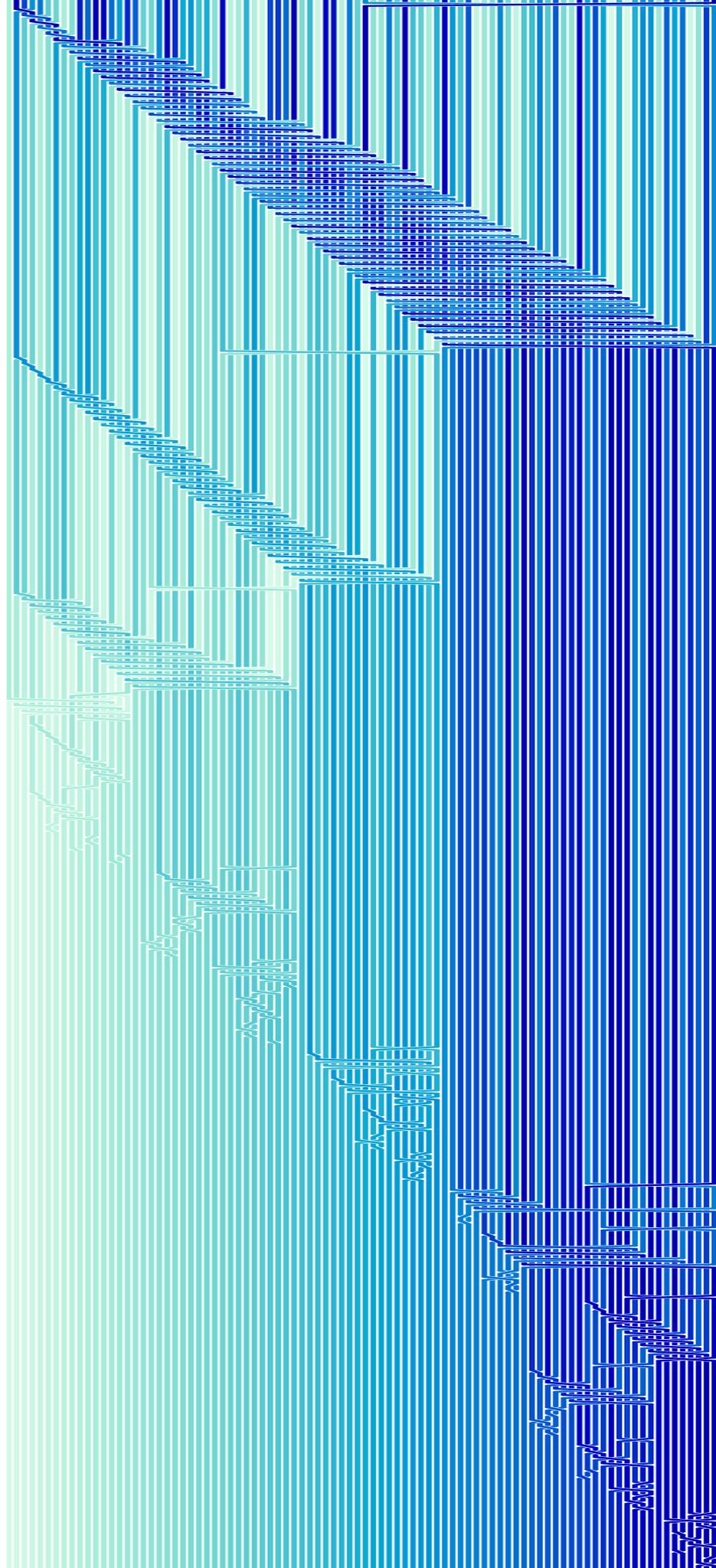
---

- Publics
    - Objectifs
- |   |  |
|---|--|
|  Architectes           |  Interpréter    |
|  Entraîneurs          |  Optimisation  |
|  Utilisateurs finaux |  Déboggage    |
|   |  Enseignement |

# Plan

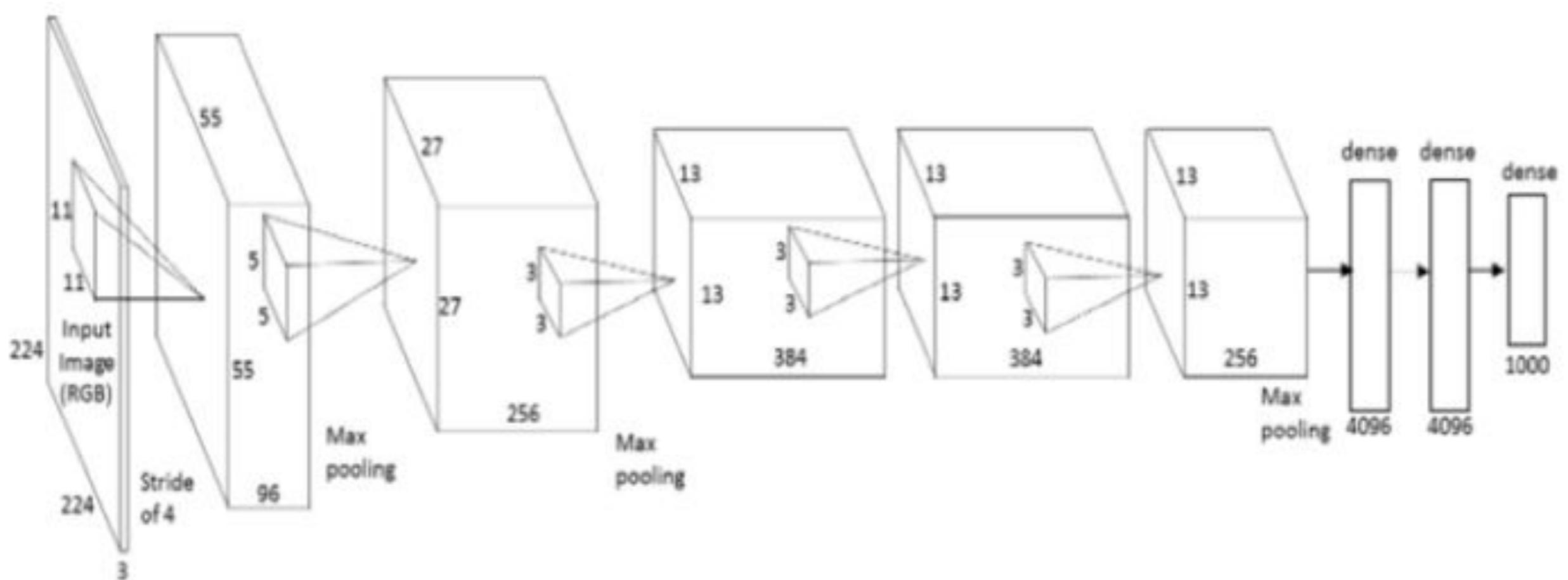
---

- Visualiser un algorithme
- Pourquoi visualiser l'apprentissage automatique ?
- Expliquer l'apprentissage automatique
  - Cas des réseaux de neurones
  - Quand ?
  - Quoi ?
  - Comment ?
- Limites

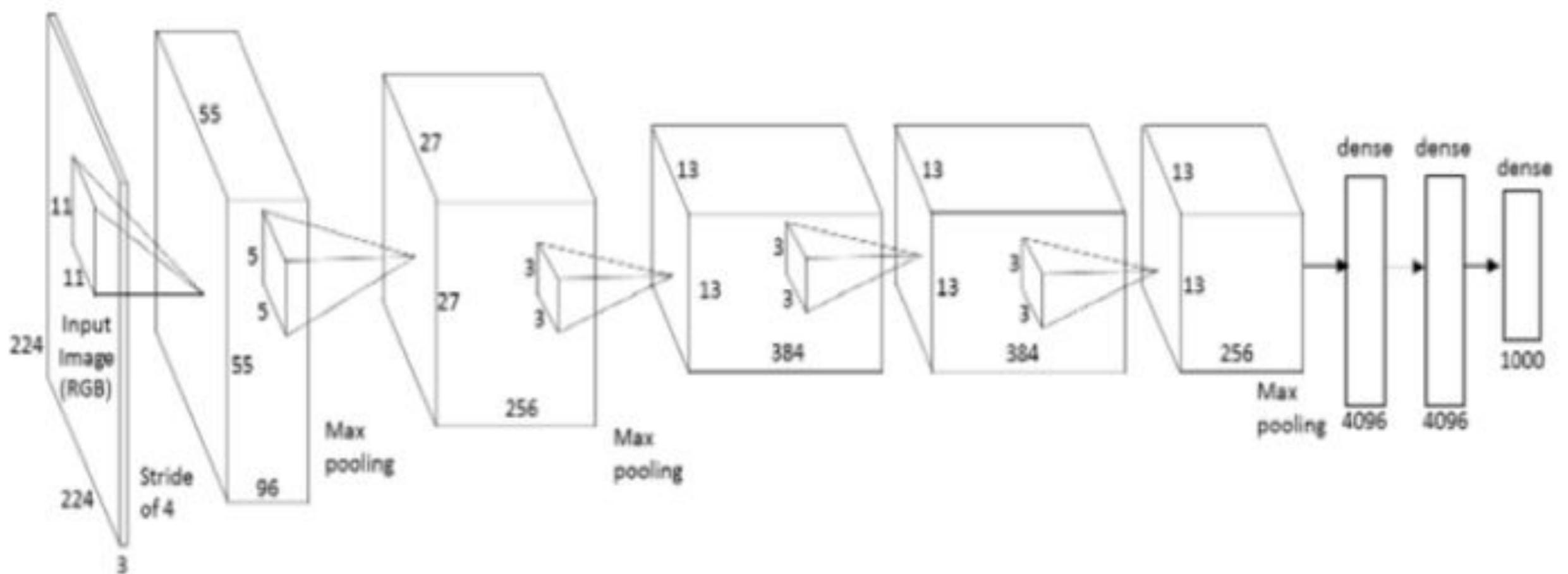


# Cas des réseaux de neurones convolutionnels

---



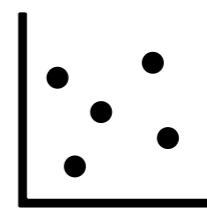
# Cas des réseaux de neurones convolutionnels



Que représenter ?

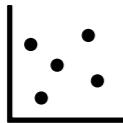


Quand ?



Comment ?

# Cas des réseaux de neurones convolutionnels

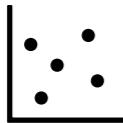
 	Noeuds & Liens	Nuage de points	Lignes & surfaces	Sorties du modèles	Représentations mixtes
Architecture du modèle					
Paramètres appris et résultats					
Étapes					
Métriques					



Entrainement

Post-entraînement

# Cas des réseaux de neurones convolutionnels

 	Noeuds & Liens	Nuage de points	Lignes & surfaces	Sorties du modèles	Représentations mixtes
Architecture du modèle					
Paramètres appris et résultats					
Étapes					
Métriques					

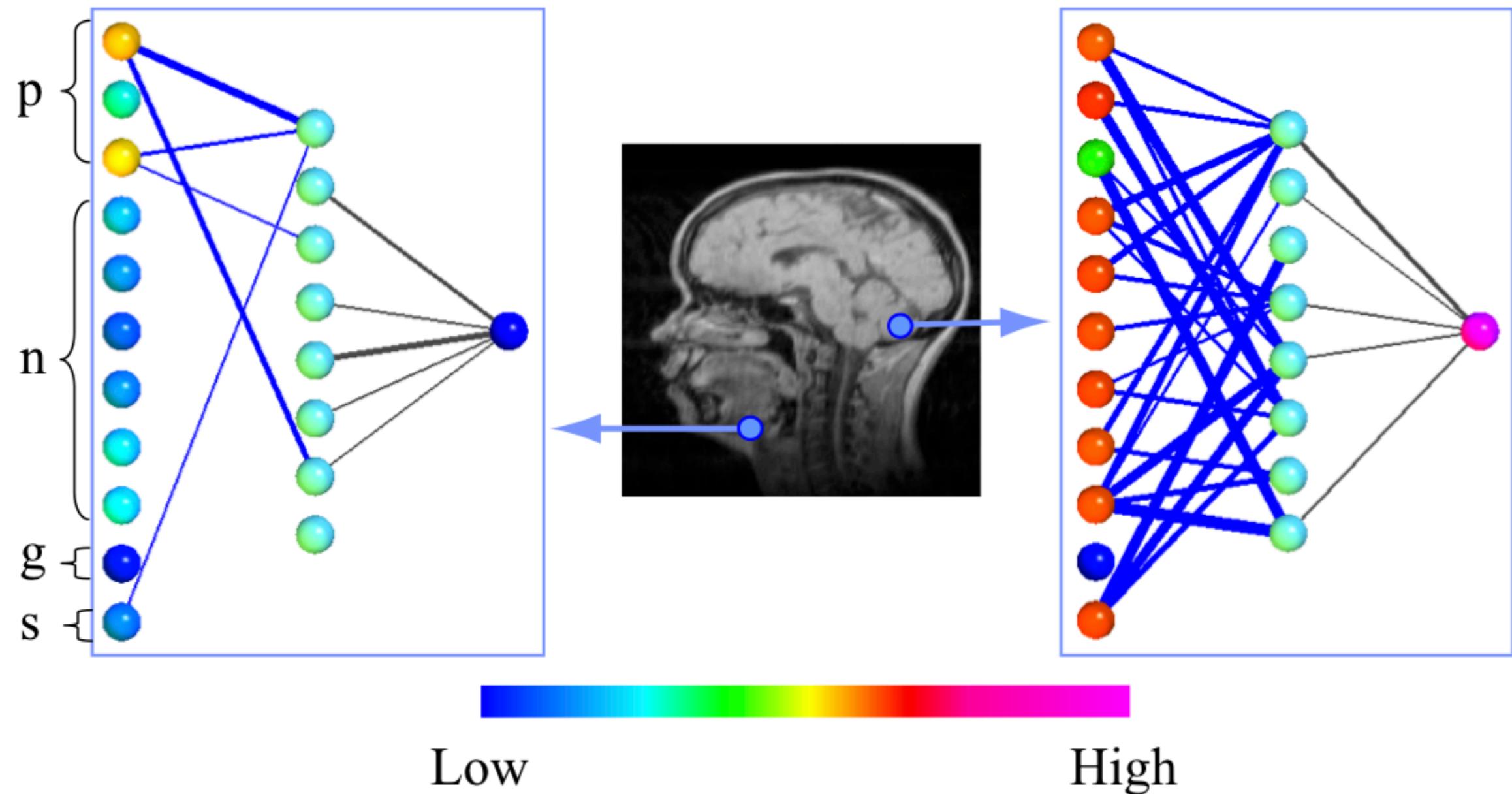


Entrainement

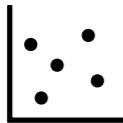


Post-entraînement

# Influence entrée - sortie



# Cas des réseaux de neurones convolutionnels

 	Noeuds & Liens	Nuage de points	Lignes & surfaces	Sorties du modèles	Représentations mixtes
Architecture du modèle					
Paramètres appris et résultats					
Étapes					
Métriques					

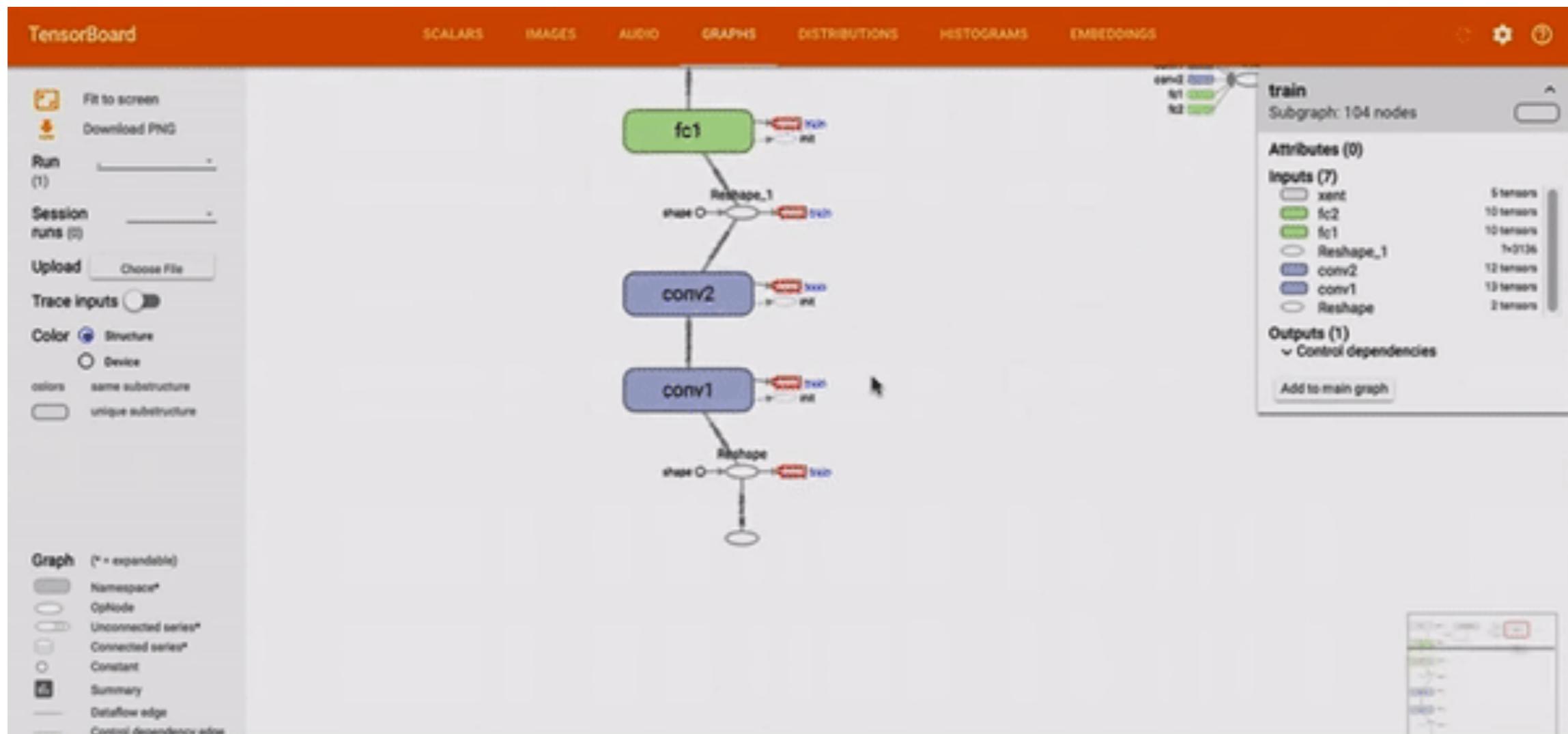


Entrainement

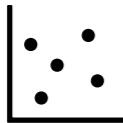


Post-entraînement

# Explorer le réseau (TensorBoard)



# Cas des réseaux de neurones convolutionnels

 	Noeuds & Liens	Nuage de points	Lignes & surfaces	Sorties du modèles	Représentations mixtes
Architecture du modèle					
Paramètres appris et résultats					
Étapes					
Métriques					

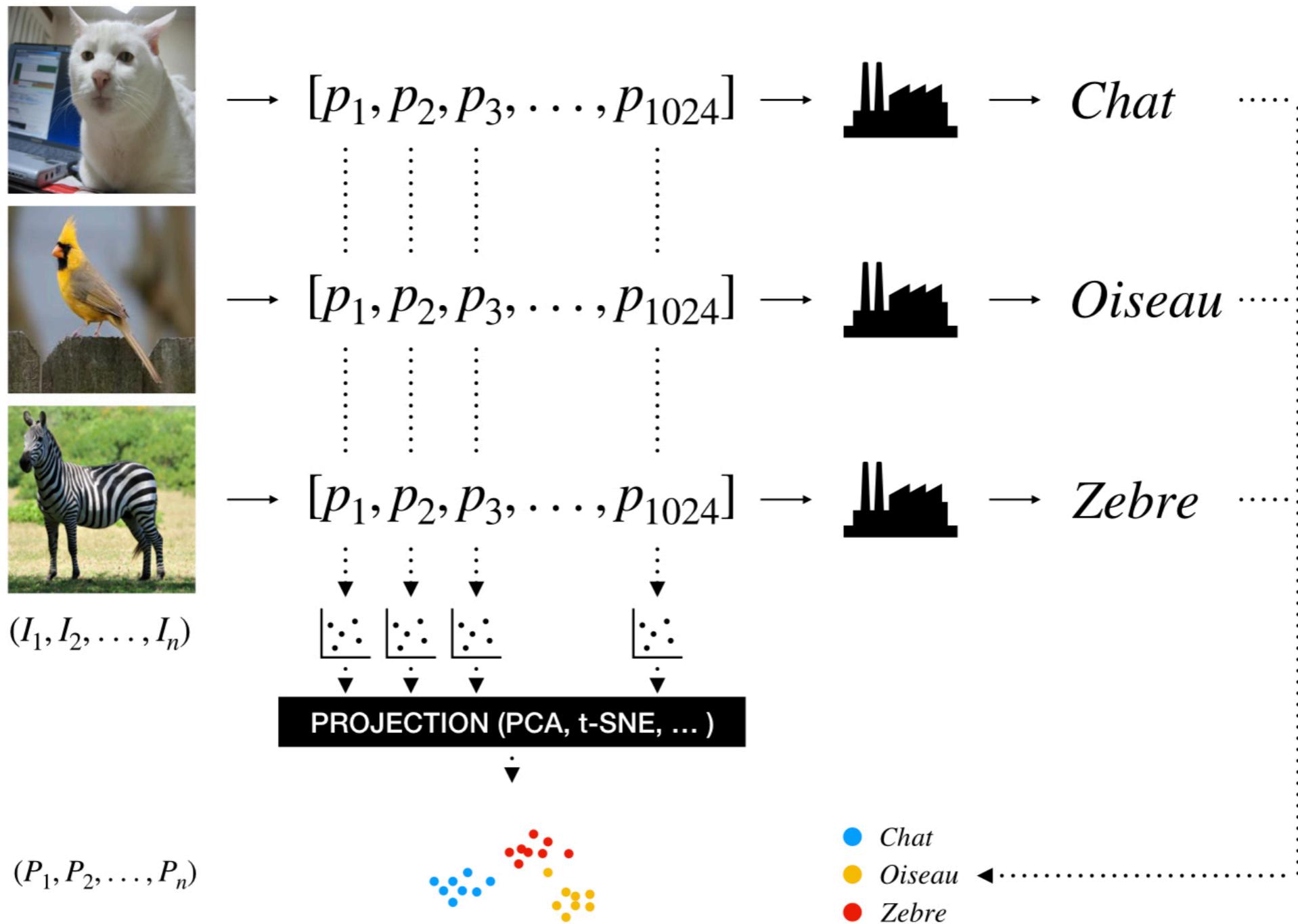


Entrainement

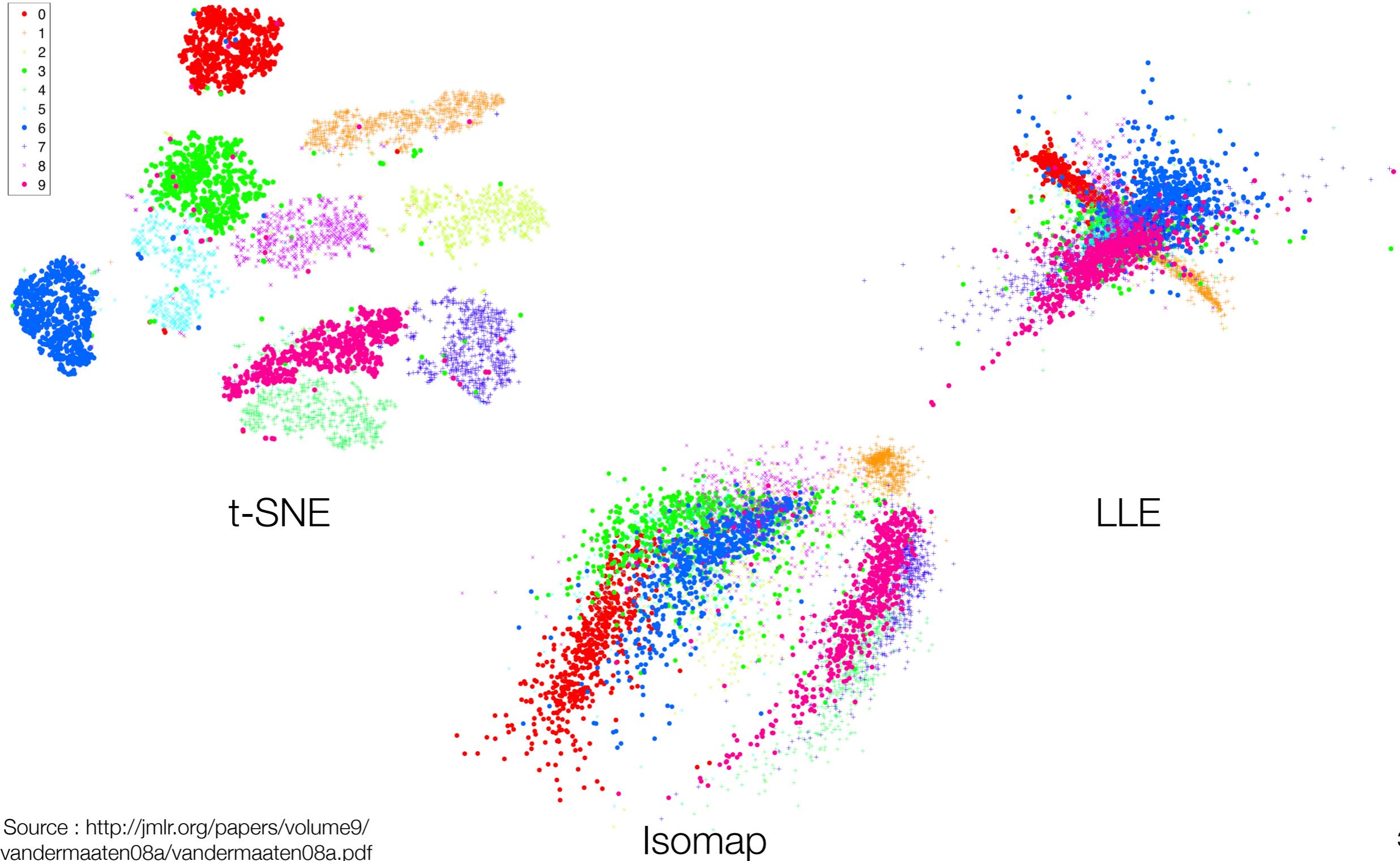


Post-entraînement

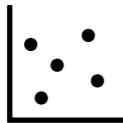
# Nuages de points : projection



# Séparation des entrées en fonction des projections



# Cas des réseaux de neurones convolutionnels

 	Noeuds & Liens	Nuage de points	Lignes & surfaces	Sorties du modèles	Représentations mixtes
Architecture du modèle					
Paramètres appris et résultats					
Étapes					
Métriques					

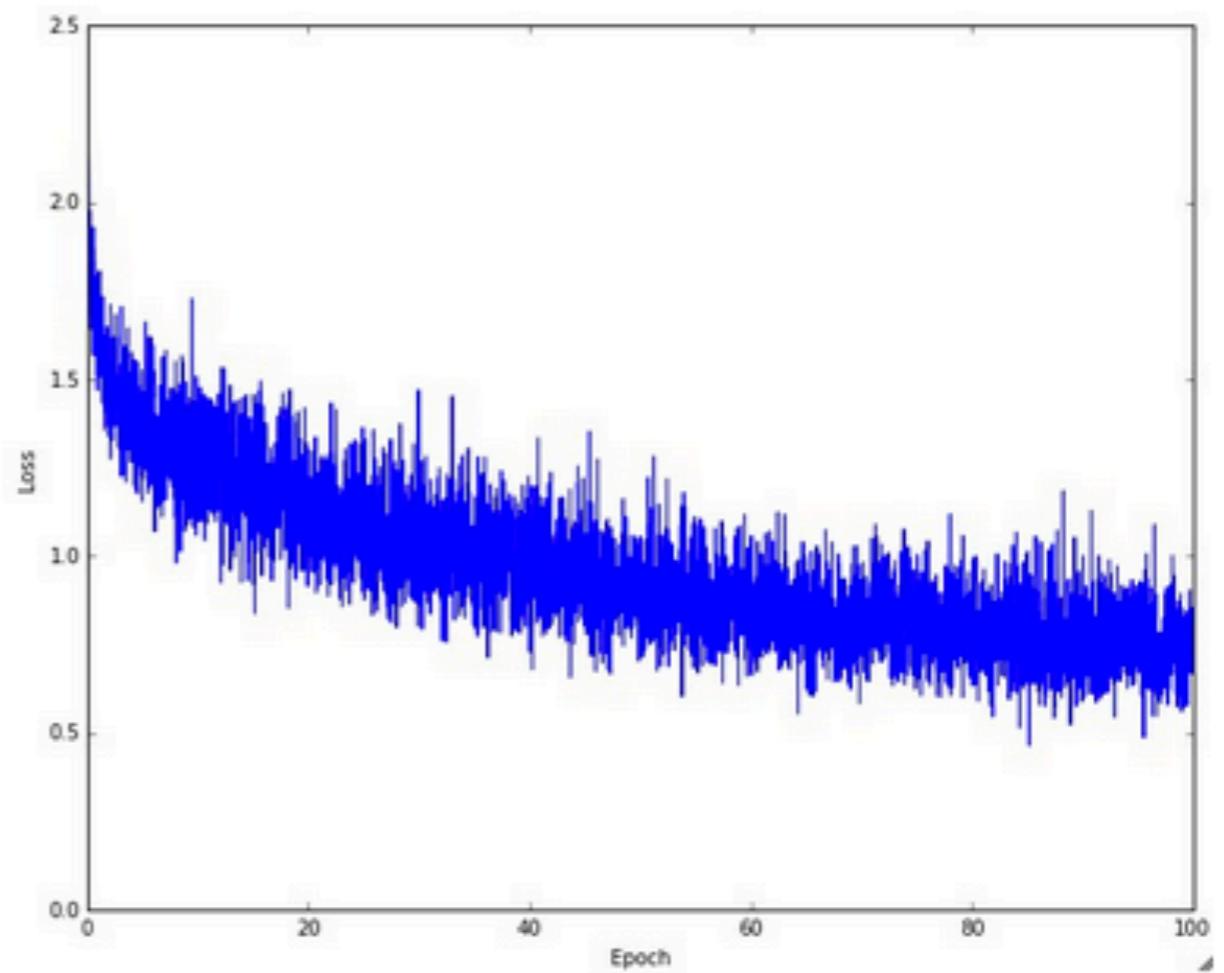
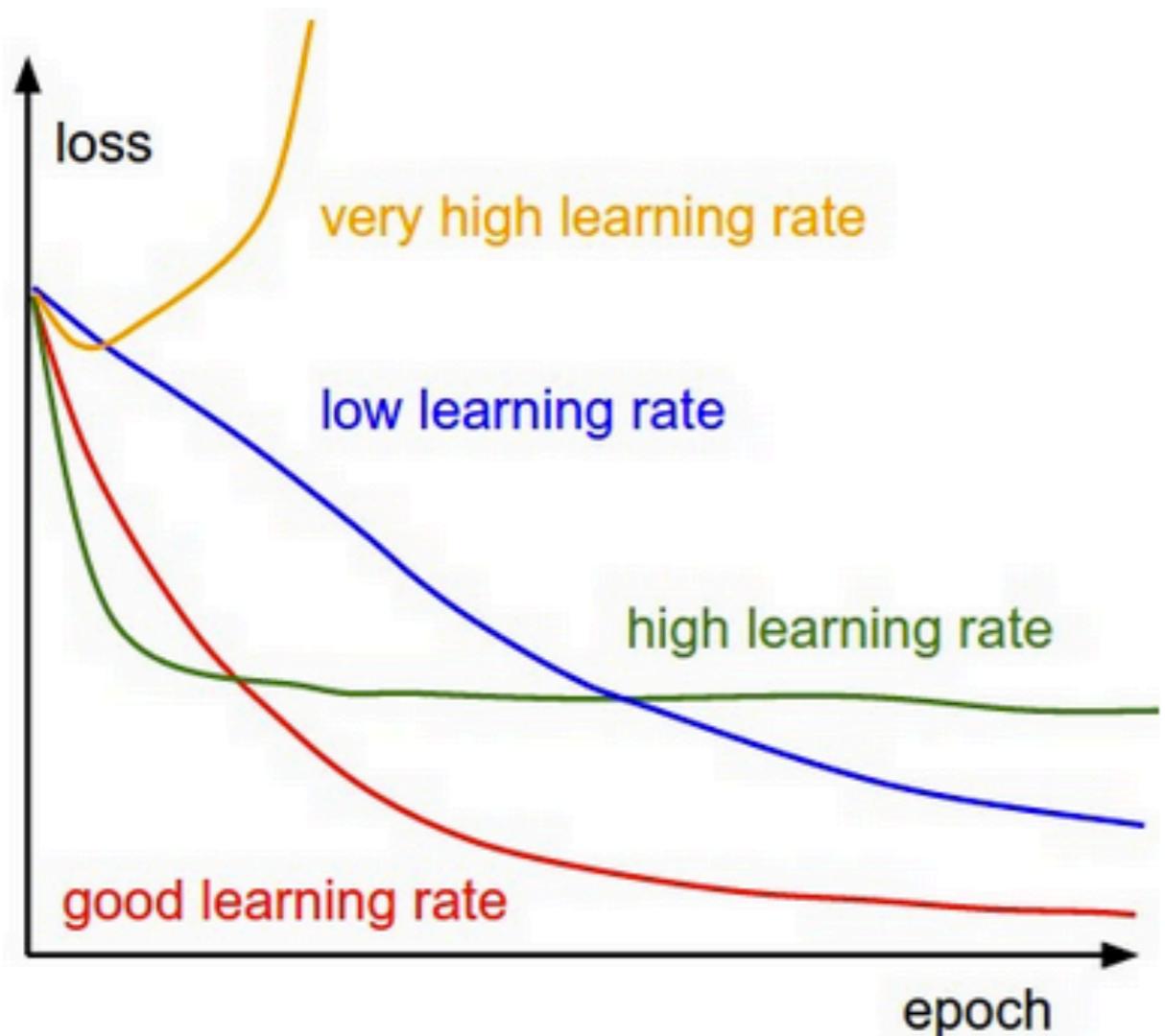


Entrainement



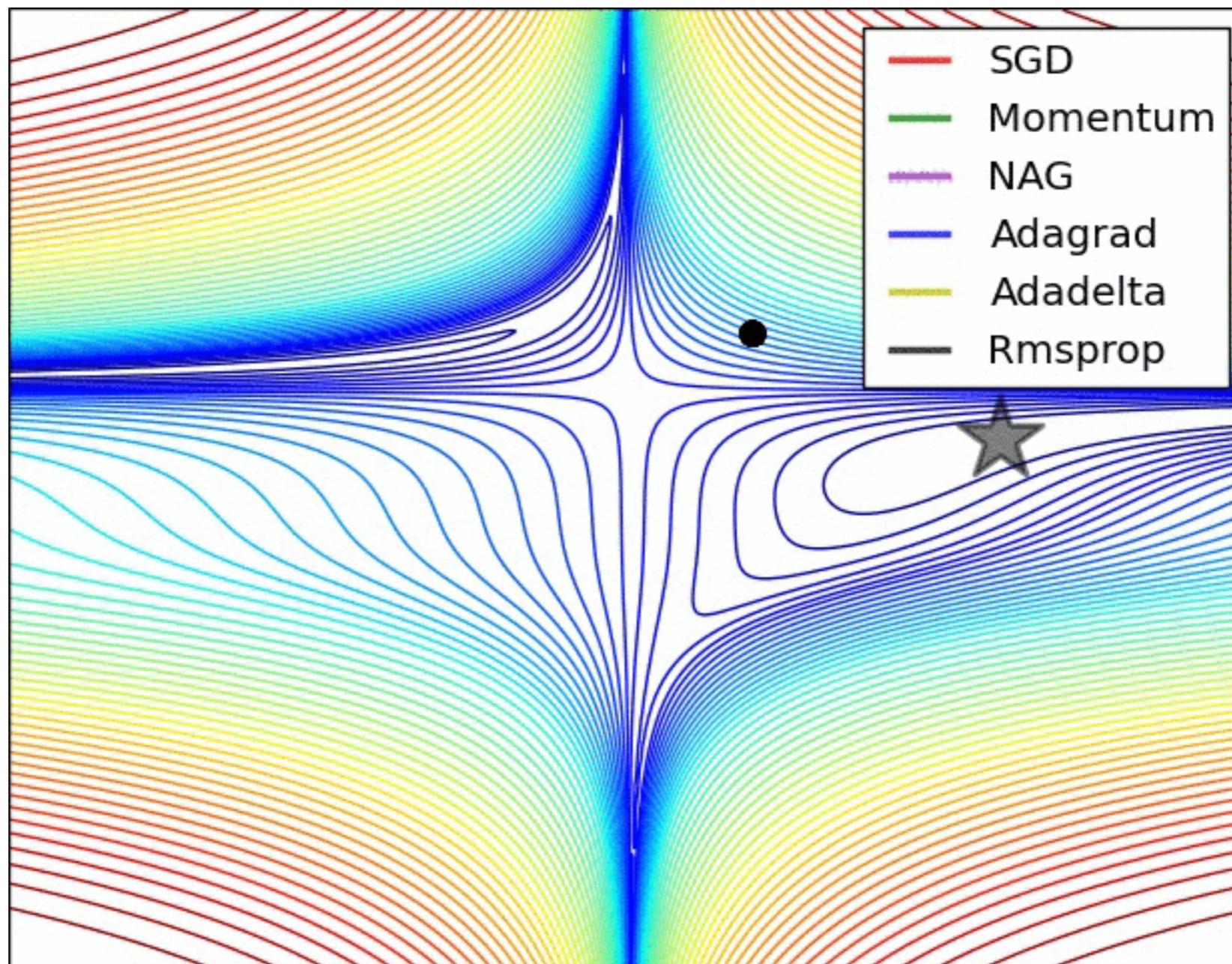
Post-entraînement

# Erreur, précision et convergence



# Erreur, précision et convergence

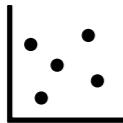
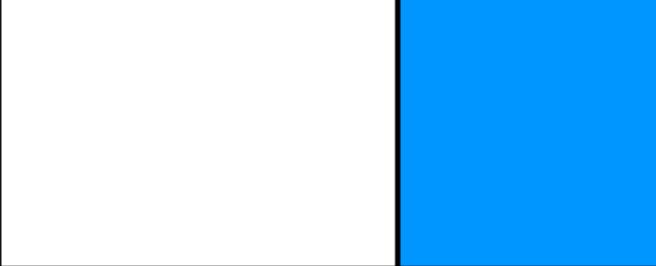
---



# Erreur, précision et convergence



# Cas des réseaux de neurones convolutionnels

 	Noeuds & Liens	Nuage de points	Lignes & surfaces	Sorties du modèles	Représentations mixtes
Architecture du modèle					
Paramètres appris et résultats					
Étapes					
Métriques					



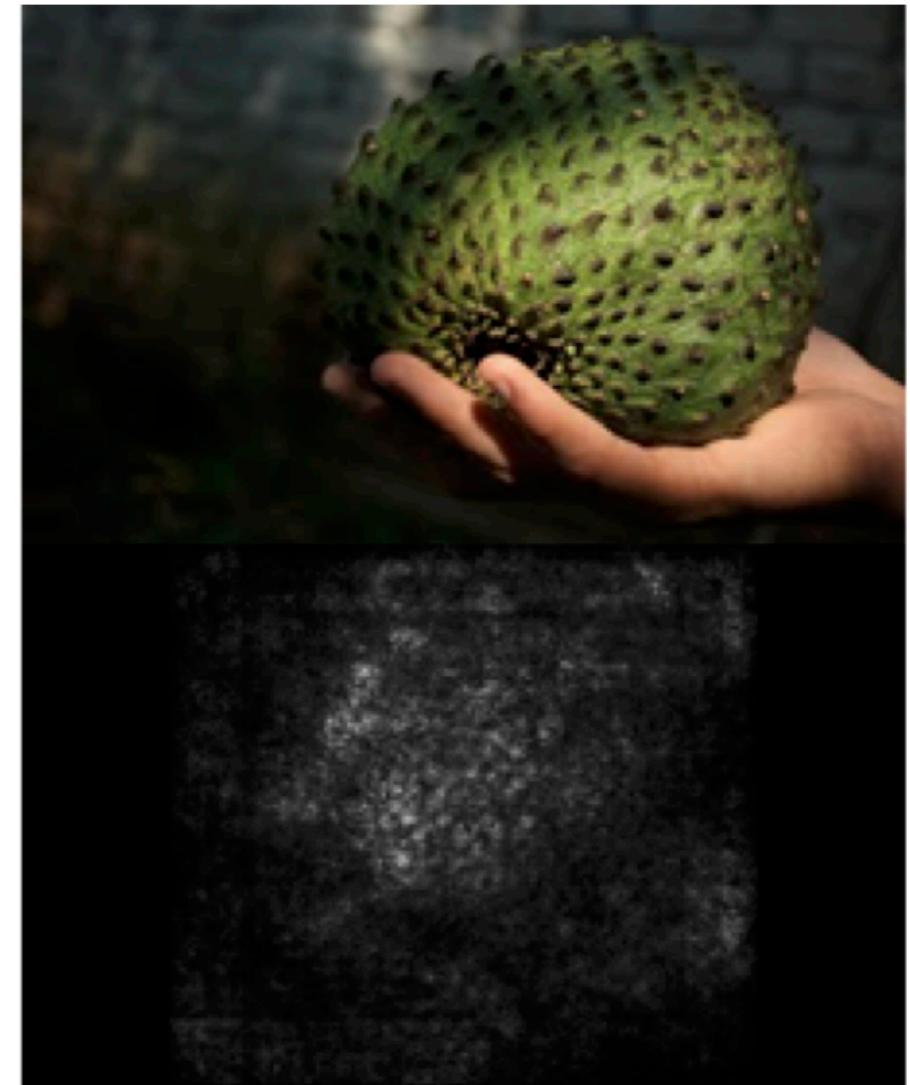
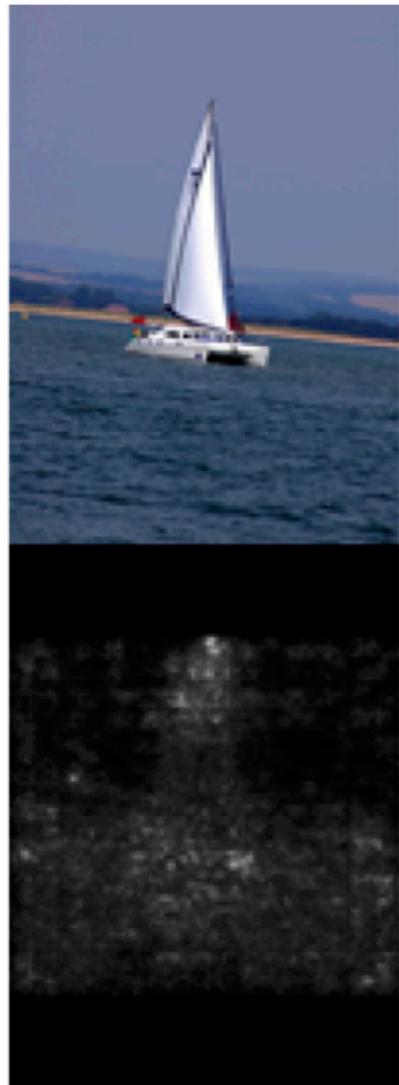
Entrainement



Post-entraînement

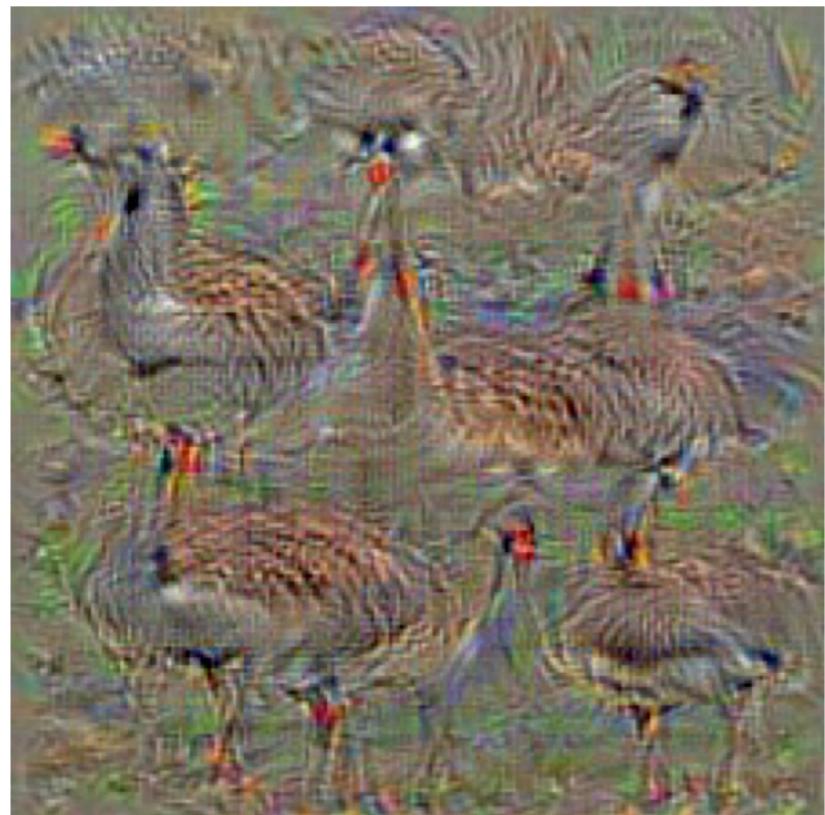
# Attribution - Saliency Maps

---



# Visualisation de classes

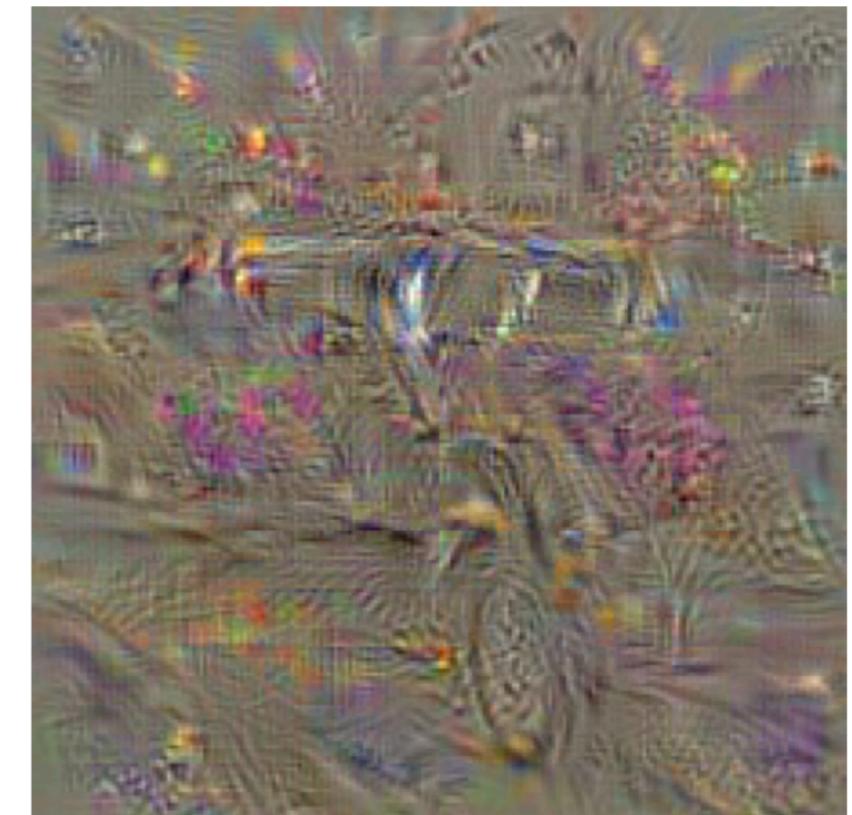
---



**goose**

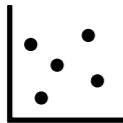
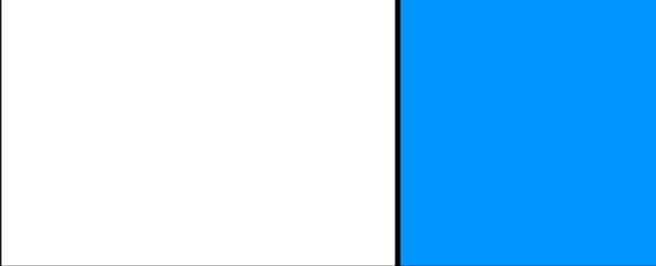


**ostrich**



**limousine**

# Cas des réseaux de neurones convolutionnels

 	Noeuds & Liens	NUAGE DE POINTS	LIGNES & SURFACES	SORTIES DU MODÈLE	REPRÉSENTATIONS INTERACTIVES
Architecture du modèle					
Paramètres appris et résultats					
Étapes					
Métriques					

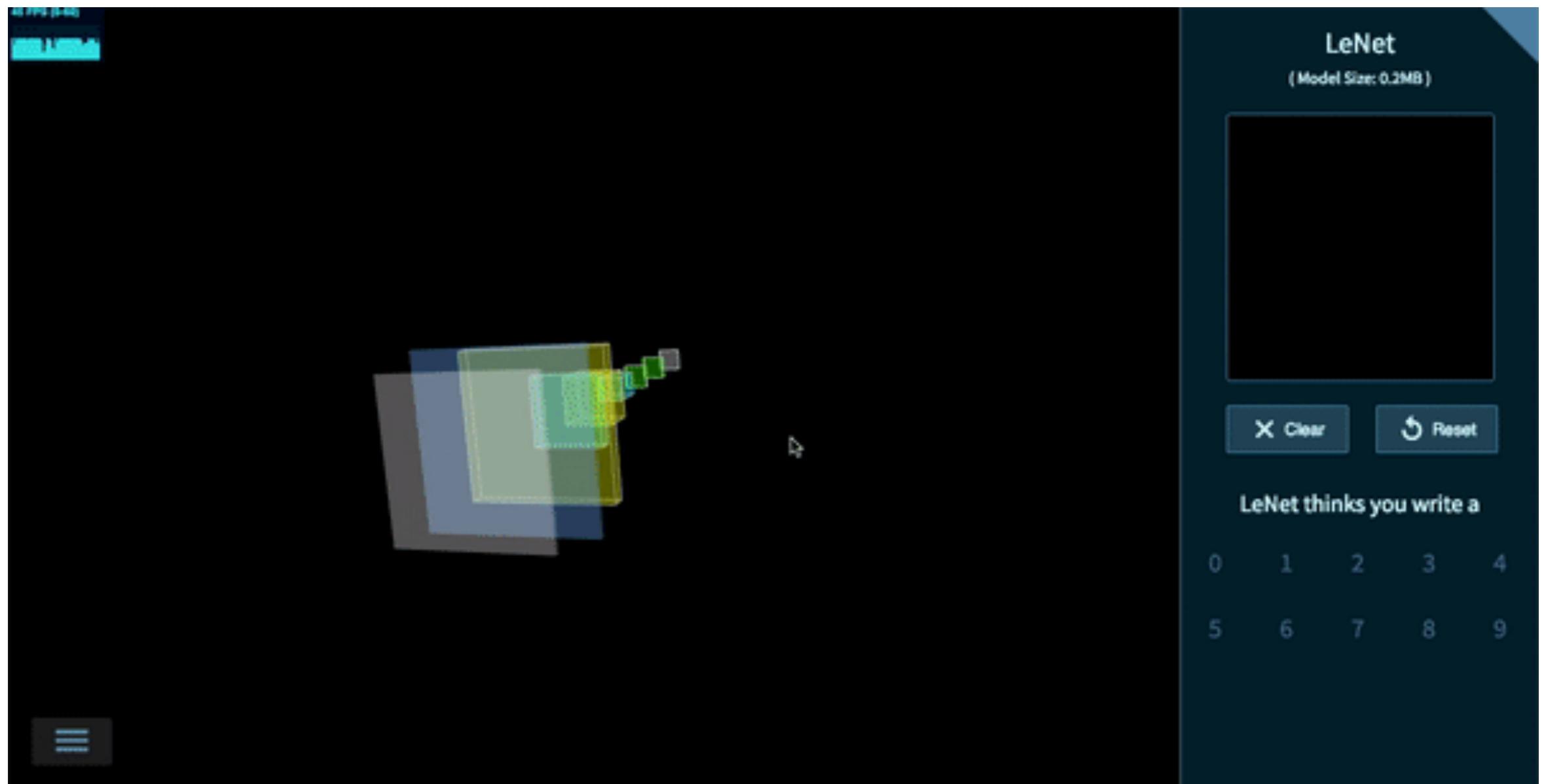


Entrainement

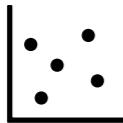
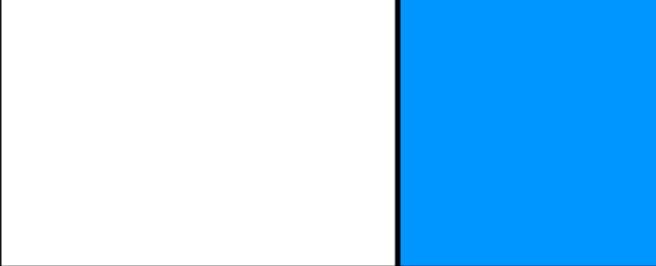


Post-entraînement

# Explorer l'architecture du réseau (TensorSpace.js)



# Cas des réseaux de neurones convolutionnels

 	Noeuds & Liens	Nuage de points	Lignes & surfaces	Sorties du modèles	Représentations interactives
Architecture du modèle					
Paramètres appris et résultats					
Étapes					
Métriques					



Entrainement



Post-entraînement

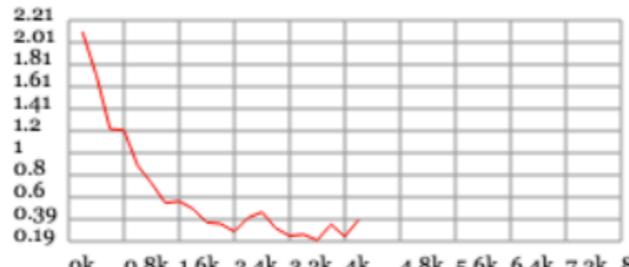
# Tableau de bord d'entraînement (ConvNetJS)

### Training Stats

pause

Forward time per example: 5ms  
Backprop time per example: 5ms  
Classification loss: 0.25308  
L2 Weight decay loss: 0.00173  
Training accuracy: 0.96  
Validation accuracy: 0.77  
Examples seen: 4383  
Learning rate: 0.01   
Momentum: 0.9   
Batch size: 20   
Weight decay: 0.001

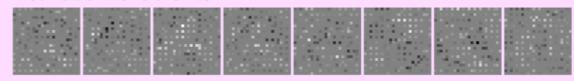
Loss:



clear graph

relu (24x24x8)  
max activation: 2.91834, min: 0  
max gradient: 0.00001, min: -0.00002

Activations: 

Activation Gradients: 

pool (12x12x8)  
pooling size 2x2, stride 2  
max activation: 2.91834, min: 0  
max gradient: 0.00001, min: -0.00002

Activations: 

Activation Gradients: 

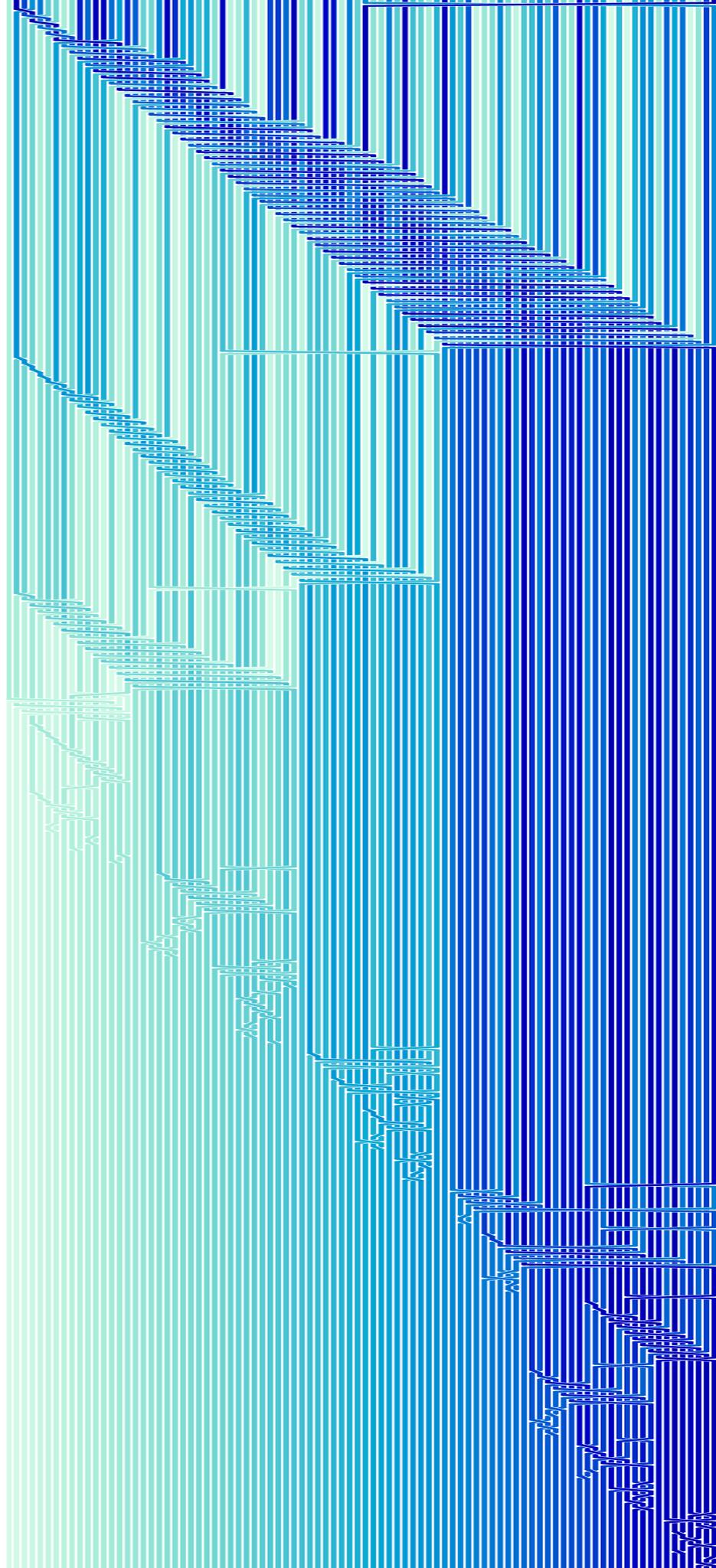
### Instantiate a Network and Trainer



# Plan

---

- Visualiser un algorithme
- Pourquoi visualiser l'apprentissage automatique ?
- Expliquer l'apprentissage automatique
- **Limites**



# Limites

---

- Trouver l'information pertinente et avoir une représentation synthétique
- Très utile pour les experts et ceux qui entraînent les modèles mais peu pour le grand public
- Long et donc coûteux
- Des compétences particulières et rares : deep learning + data visualisation

# Merci

Remerciements : Théo Jaunet