

# IA

1. Cadrage historique
2. Apprentissage... supervisé, auto supervisé, non supervisé...
3. Principe des réseaux de neurones
4. Succès (ou non) actuels dans la tech
5. Génération de texte
6. Biais
7. Apprentissage de la programmation à l'Université et LLMs

Laurent.Signac@univ-poitiers.fr

Ressources : [https://ensip.gitlab.io/pages-info/cours/formations/ia\\_2023/](https://ensip.gitlab.io/pages-info/cours/formations/ia_2023/)

# Cadrage historique

Objectif :

Réaliser avec des machines des tâches traditionnellement réalisées par des humains, et dont on *estime* qu'elles nécessitent de l'intelligence

Théorie triarchique de l'intelligence (Sternberg 1985) :

- intelligence analytique ;
- intelligence créative (dont émotionnelle) ;
- intelligence pratique ( $\approx$  débrouillardise).

## (Très) brève histoire de l'IA

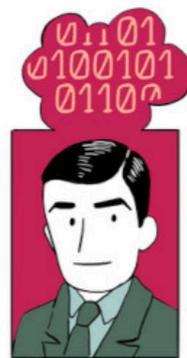
- Naissance de la **cybernétique** en 1943, avec le **neurone formel** de Mc Culloch et Pitts
- Naissance du terme **intelligence artificielle** en 1956 lors d'une conférence historique dans le New Hampshire
- Perceptron de F. Rosenblatt en 1958
- Jusqu'en 1970 l'**approche logique** a le vent en poupe, mais l'**apprentissage automatique** se développe peu à peu
- Premier **hiver de l'IA** jusqu'en 1980
- Années 80,  **systèmes experts** (approche logique) et **réseaux de neurones** (apprentissage automatique), **rétropropagation du gradient**
- Second **hiver de l'IA** ( $\approx$  80-90)
- Depuis le début du millénaire, les deux approches (logique et apprentissage automatique) coexistent, mais les **réseaux de neurones, depuis 2010, sont les stars de l'IA**
- 2010 : On dispose de suffisamment d'**exemples** (base d'apprentissage) et de **puissance** de calcul pour que ça marche

# A.I. HISTORICAL TIMELINE



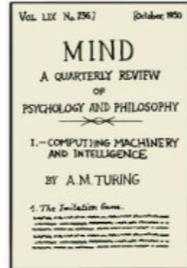
**Ada Lovelace**  
First design of a computer program by Ada Lovelace

1837



Invention of the algorithm concept (Turing machine)

1939



Test for machines' "intelligence" (Turing test)

1950



**Margaret Masterman**  
Creation of the Cambridge Language Research Unit

1953



First psychotherapist chatbot (ELIZA)

1965



**MYCIN**  
Expert system used for the diagnosis and therapy of infectious diseases (MYCIN)

1972



First general-purpose mobile robot (Shakey the robot)

1966



First artificial intelligence program (Logic Theorist)

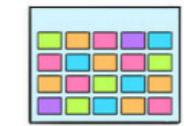
1956



1<sup>st</sup> AI winter  
1974-1980

— ImageNet at the origin of major advances in image recognition —

## IMAGE NET



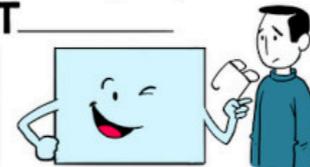
Large-scale hierarchical Image Database (ImageNET)

2009



DeepLearning in Image recognition (AlexNet)

2012



Computer vision exceeds human vision

2015



Chess-playing expert system (Deep Blue)

1997



AI4T

1921

Creation of the word "ROBOT"

**RUR**



1945

Electronic Numerical Integrator Analyser and Computer (ENIAC)



1954

A demonstration of machine translation: Georgetown-IBM Experiment



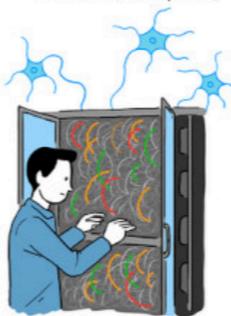
1956

Birth of the term "Artificial Intelligence"



1957

First single layer neural network (Perceptron)



General Problem Solver



1986

The computer is given a voice (NETalk)



First computer controlled vehicles (Navlab)



AI boom  
Rise of expert systems

1980-1987

2<sup>nd</sup> winter of AI

1988-1993

2013

Natural language processing technique (Word2Vec)



2018

Turing Award for recent advances in deep learning



2015

Announcement of fully autonomous cars



# Apprentissage automatique



- Programme explicite (programmation classique) : complexe et (parfois) laborieux
- Programme implicite : calcul d'**optimisation** automatique faisant **converger** un **modèle** sur un **jeu de données**.
  - Le programme implicite contient :
    - partie fixe : type de modèle (réseaux de neurones, régression, arbre...), architecture...
    - partie variable : paramètres du modèle (ce qu'on optimise)
  - le calcul d'optimisation prend des formes variées

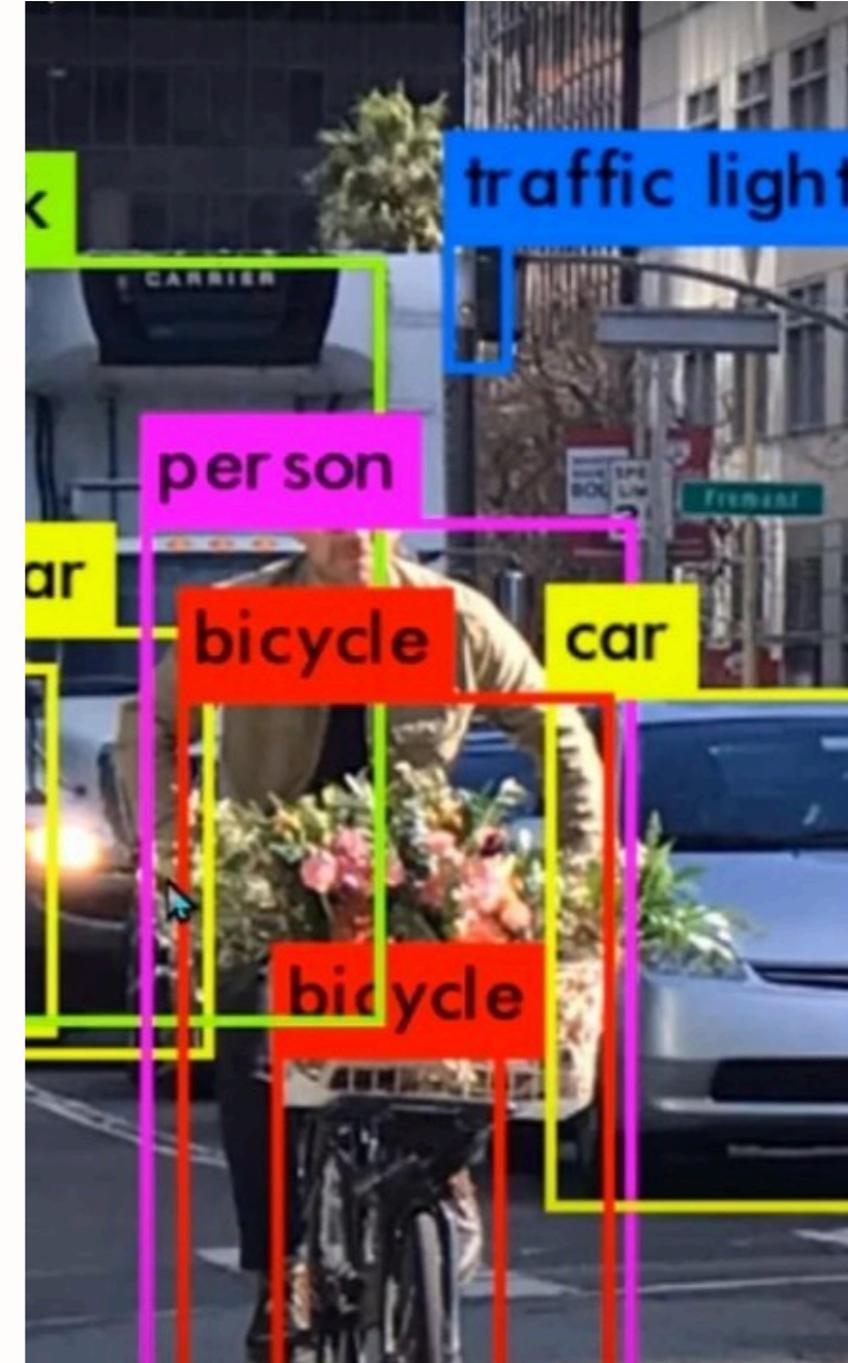
**Apprendre** : trouver les meilleurs valeurs possibles pour les paramètres / réaliser un calcul d'optimisation

## Apprentissage supervisé

On fournit des exemples d'entrées / sorties souhaitées : (image de chat ou de chien en entrée, tag chat ou chien en sortie).

Exemples de tâches :

- Identifier un code postal manuscrit
- Dépister des tumeurs sur des images médicales
- Reconnaître des visages
- Taguer une scène



## Apprentissage non supervisé

- on ne connaît pas les sorties désirées
- on ne fournit que des entrées
- on attend du système qu'il fournisse des sorties pertinentes (partitionnement par exemple).

Exemple de tâche : découvrir des profils de consommateur à partir de tickets de caisse.



# Apprentissage auto supervisé

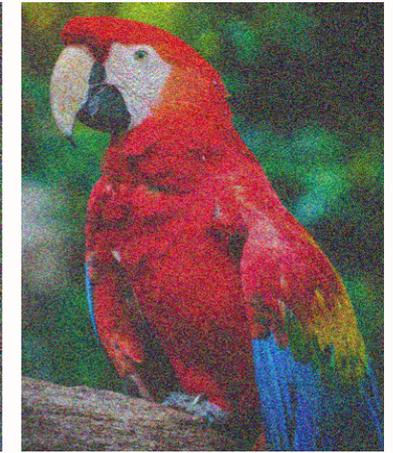
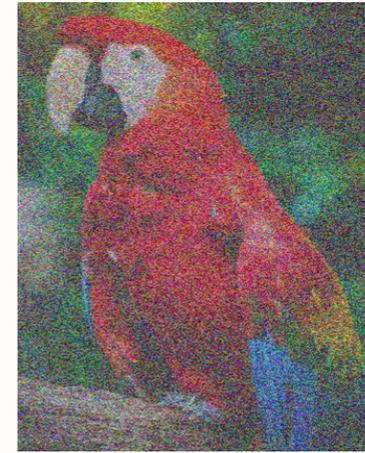
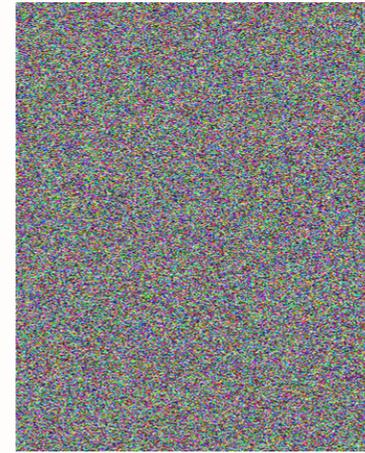
- on fait l'économie de l'étiquetage
- très utilisé pour les IA génératives

L'intelligence artificielle générative ou IA générative (ou *GenAI*) est un type de système

capable de générer du **contenu**, des **images** ou d'autres médias en réponse à des invites (ou "prompt")<sup>1,2</sup>. Les modèles génératifs apprennent les modèles et la structure des **données**, puis génèrent **des données** similaire aux **données** mais avec un certain degré de nouveauté (plutôt que de simplement classer ou prédire les données)<sup>3</sup>.

L'IA **générative** peut être *unimodale* ou *multimodale*; les systèmes unimodaux n'acceptent qu'un seul type **de données** (par exemple, du texte), tandis que les systèmes **multimodaux** peuvent accepter plusieurs types **de données** (par exemple, du texte et des images)<sup>4</sup>.

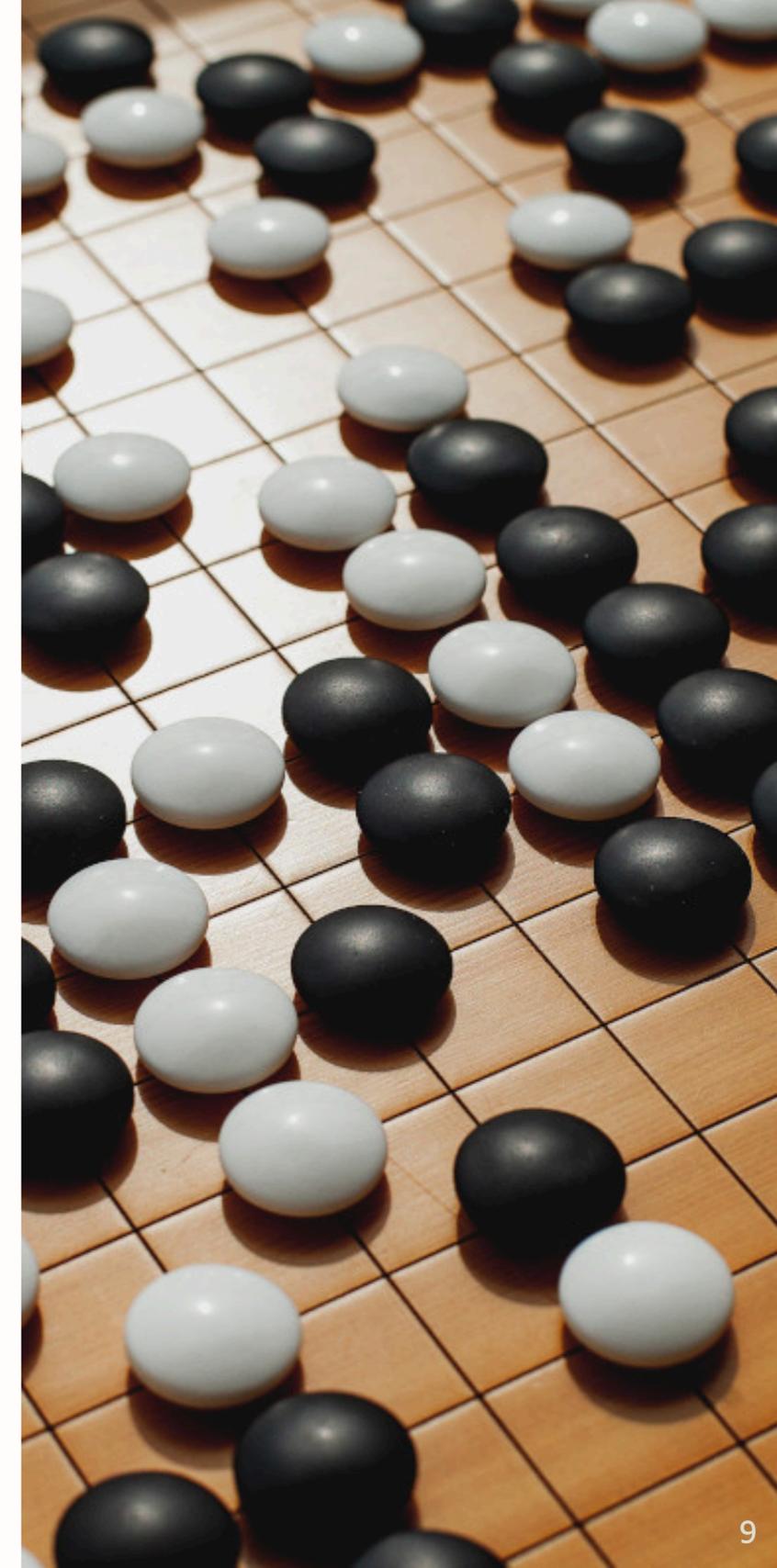
Les cadres les plus importants pour aborder l'IA générative comprennent **l'auto-supervision** et **l'entraînement par renforcement**<sup>5,6</sup>. Les **GANs** se composent de deux parties : un réseau générateur qui crée de nouveaux échantillons de données et un réseau discriminatoire qui évalue si les échantillons sont réels ou faux. Les deux réseaux sont formés ensemble dans le cadre d'un processus concurrentiel, le réseau générateur essayant continuellement de produire des échantillons de meilleure qualité et plus réalistes, tandis que le réseau discriminatoire s'efforce d'identifier avec précision les faux échantillons. Les **GANs** sont des réseaux de neurones artificiels fondés sur l'architecture du **réseau de neurones convolutifs** pré-entraînés sur de grands ensembles de données de texte non étiqueté et capables de générer **des données** de type humain<sup>7,8</sup>. Ils utilisent **l'auto-supervision** (abrégé **GAN** de l'anglais **Generative Adversarial Network**) pour produire des données fondées sur l'ensemble de données d'entraînement qui a été utilisé pour les créer<sup>9</sup>.



## Apprentissage par renforcement

- machine plongée (virtuellement ou non) dans un environnement
- reçoit un **feedback** sur ses actions.

La quantité d'expériences que doit tenter une machine avant d'avoir un comportement intéressant est considérable.



# Multitude de problèmes et types de modèles ?

## Méthodes

- régressions (linéaires ou non)
- k plus proches voisins
- arbres de décision
- réseaux de neurones (dont **deep learning**)
- réduction de dimension (analyse en composantes principales...)
- clustering (cartes de kohonen, k-means...)
- ...

## Problèmes

- classification (sortie discrète)
  - binaire
  - multi classe (qui s'excluent mutuellement ou non)
- régression (sortie continue)
- génération

# Principe des réseaux de neurones

Comparaison table de mixage dûe à G.P (Tech Café).

1 neurone	1 table de mixage
1 réseau	plusieurs tables de mixage
paramètres	boutons réglables
apprendre	régler tous les boutons
hyperparamètres	type de table, et manière de les relier

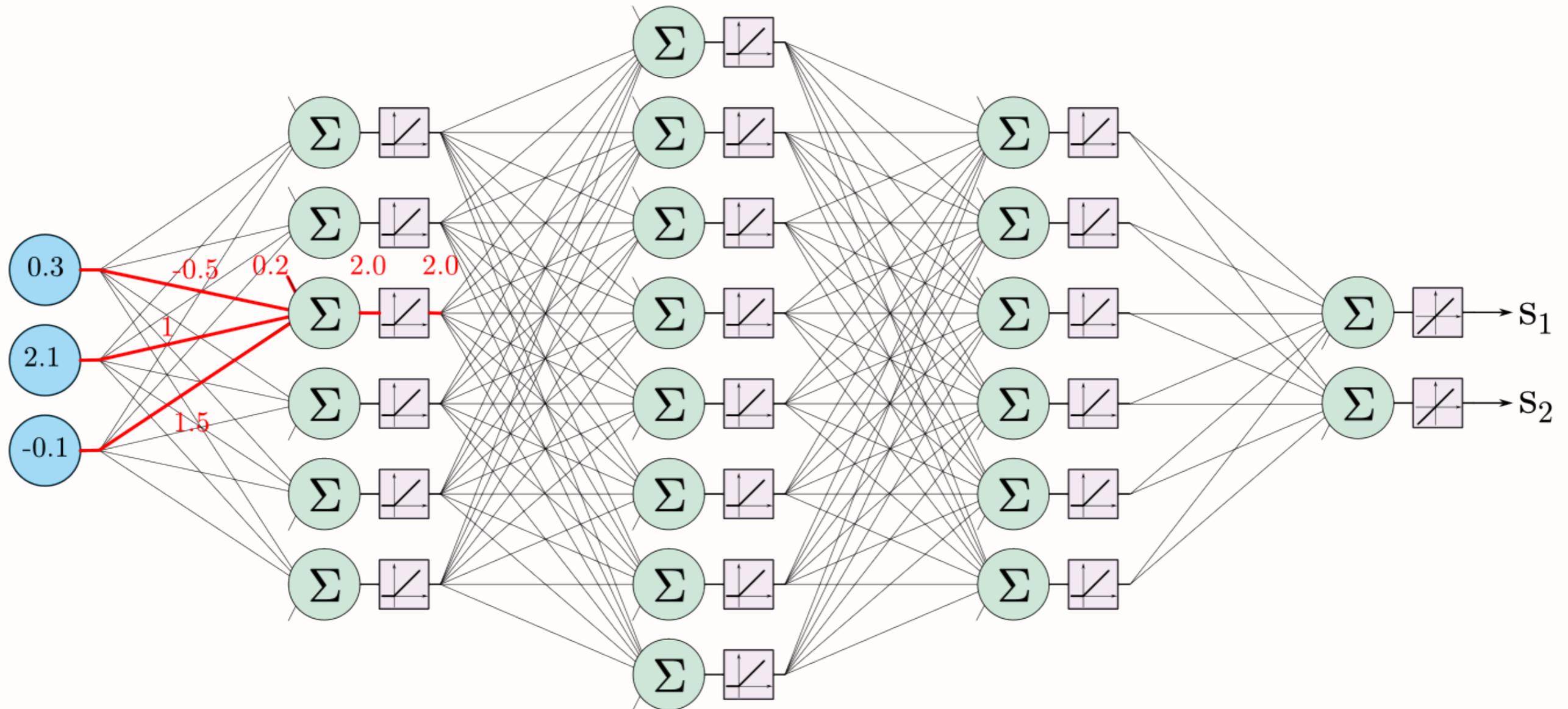


## Tout est nombre

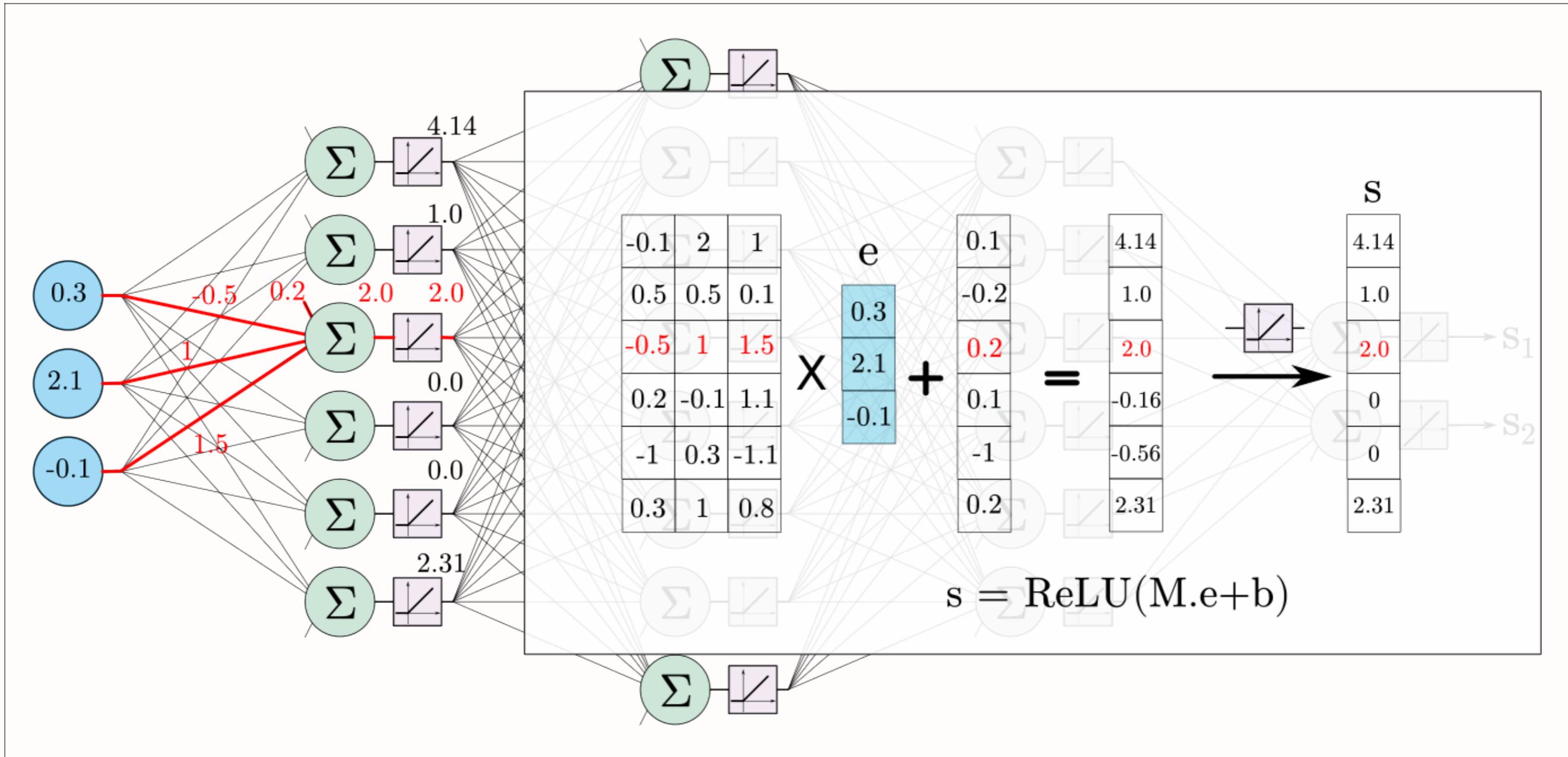
- pixels d'une image, mots d'un texte  
→ nombres.
- tous les réseaux de neurones font du calcul tensoriel sur des nombres.



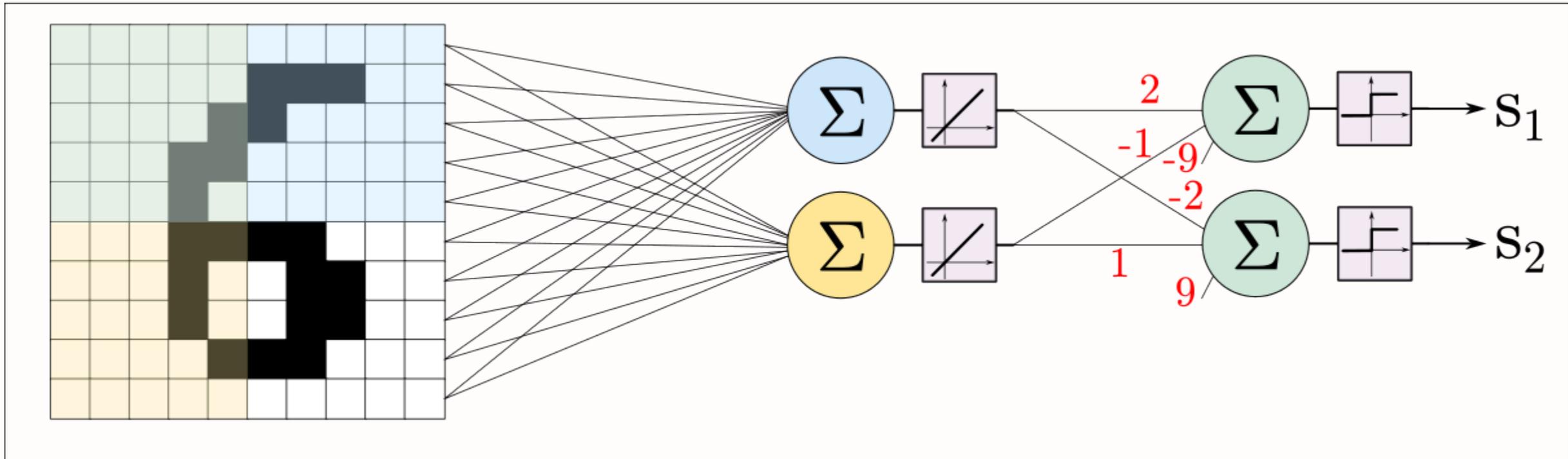
# Opérations matricielles (1/2)



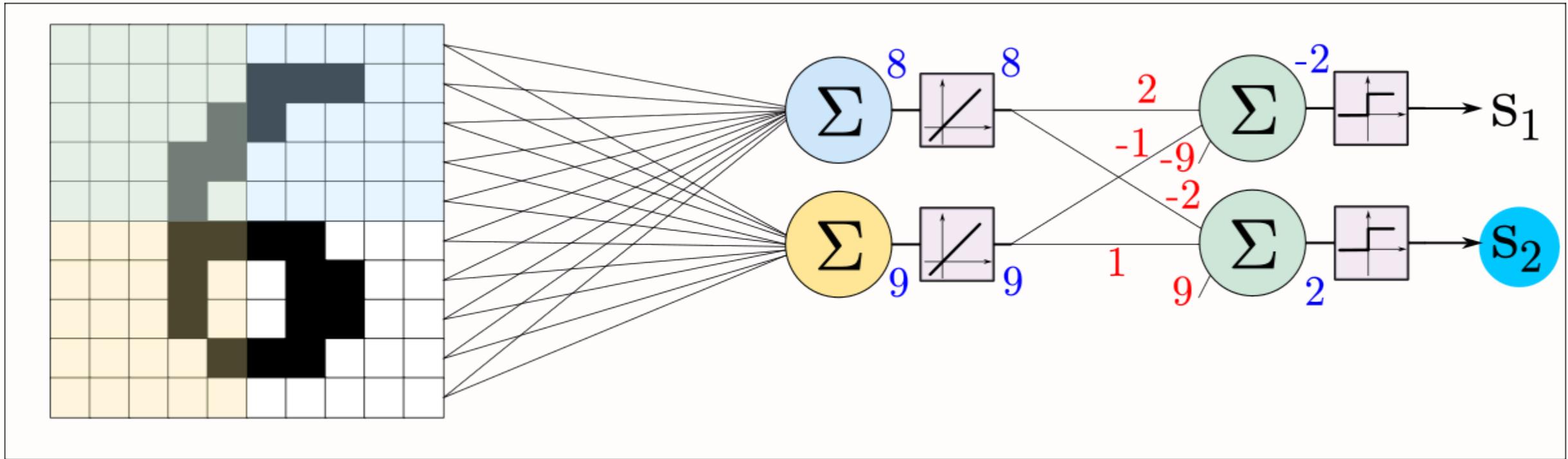
## Opérations matricielles (2/2)



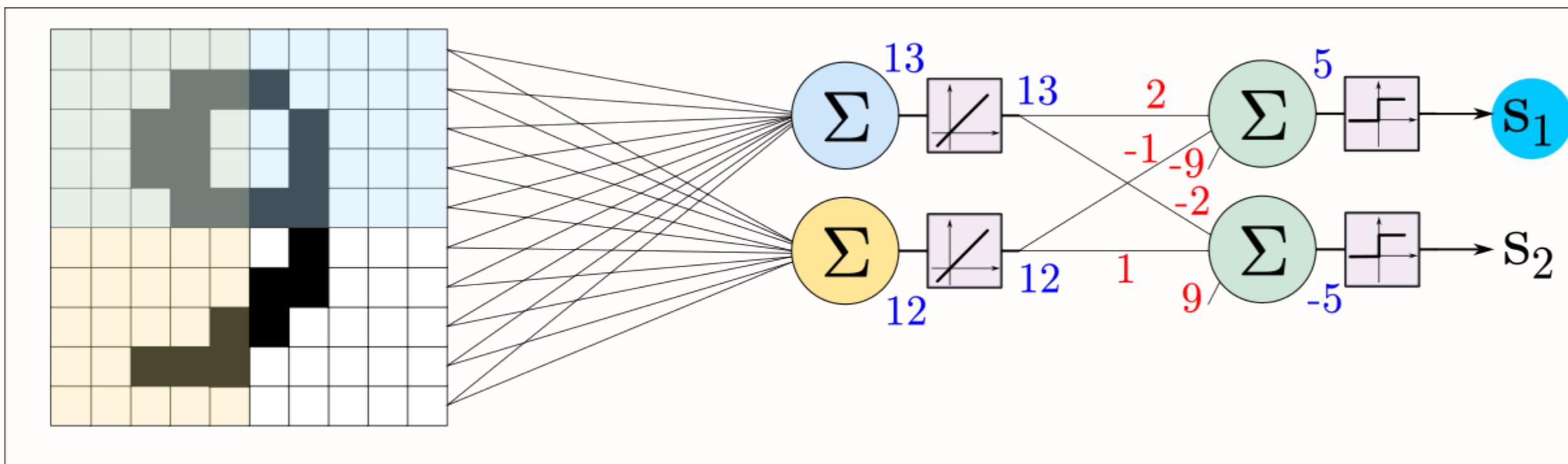
## Exemple complet 1/4 Entrez dans la tête d'une IA



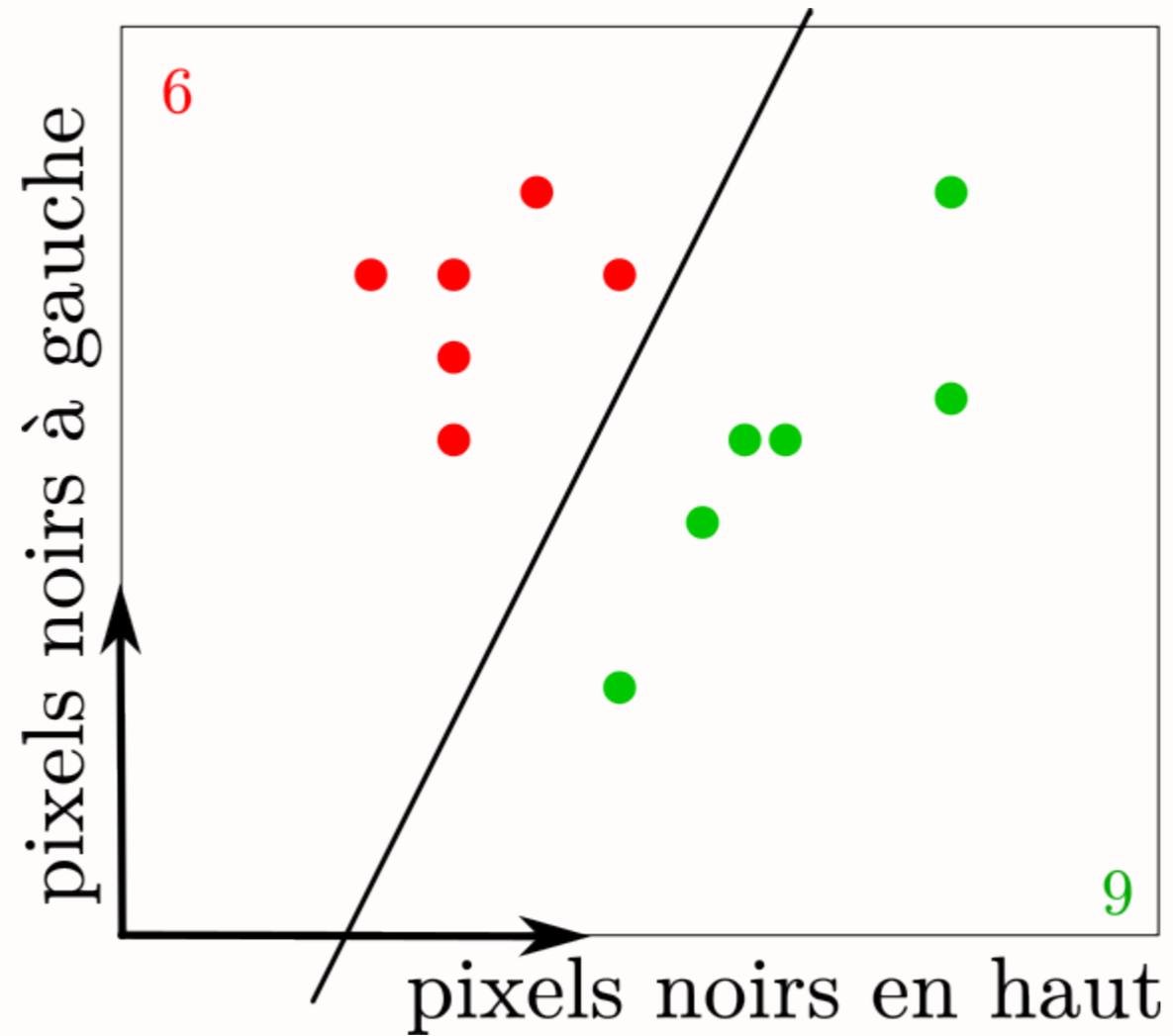
## Exemple complet 2/4



## Exemple complet 3/4

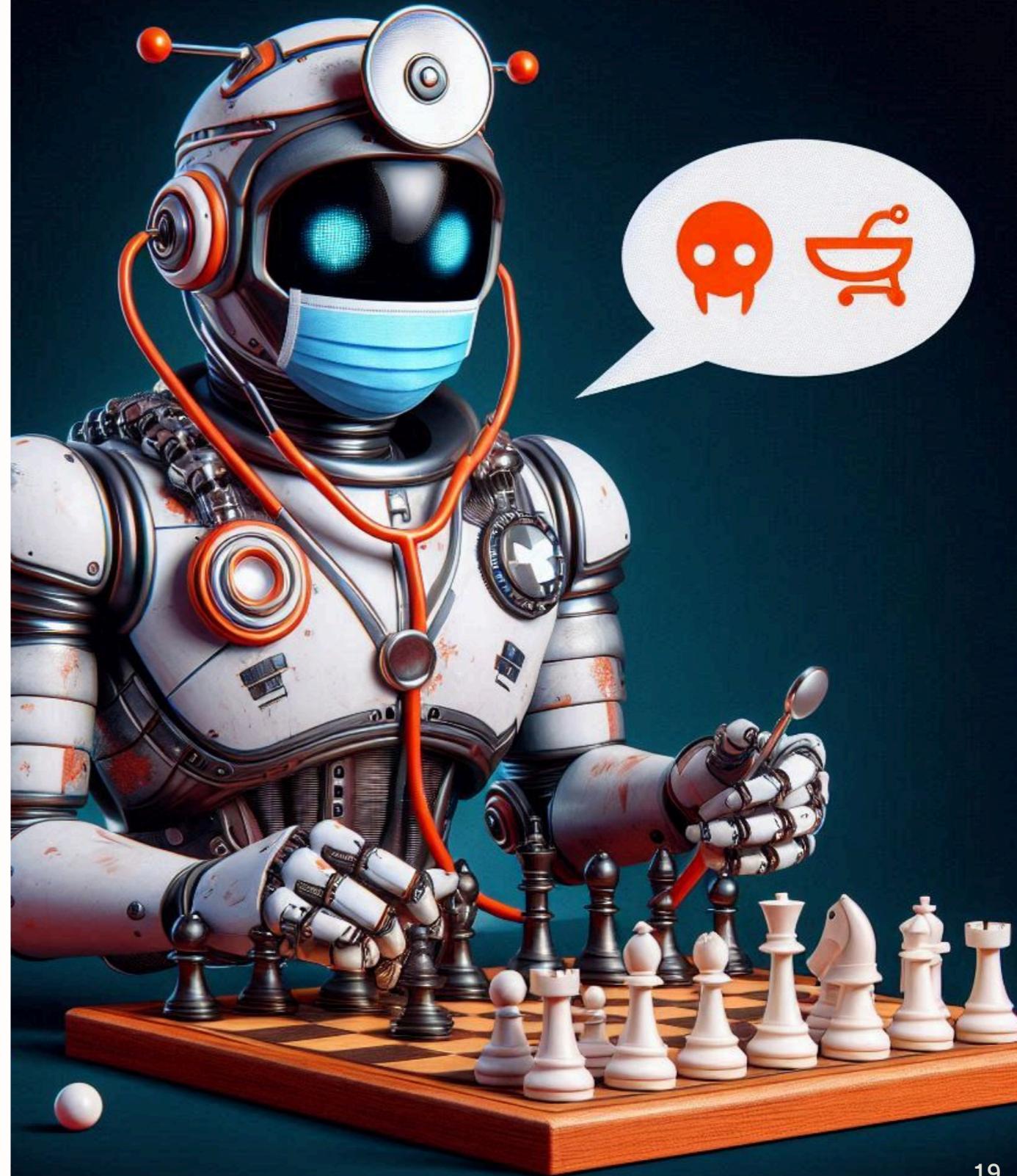


## Exemple complet 4/4



## Succès...

- Jeux (échecs, go...)
- Assistants vocaux
- Traduction automatique
- Analyse d'images médicales
- Restauration / génération d'images et de vidéo
- Les LLMs (agents conversationnels)

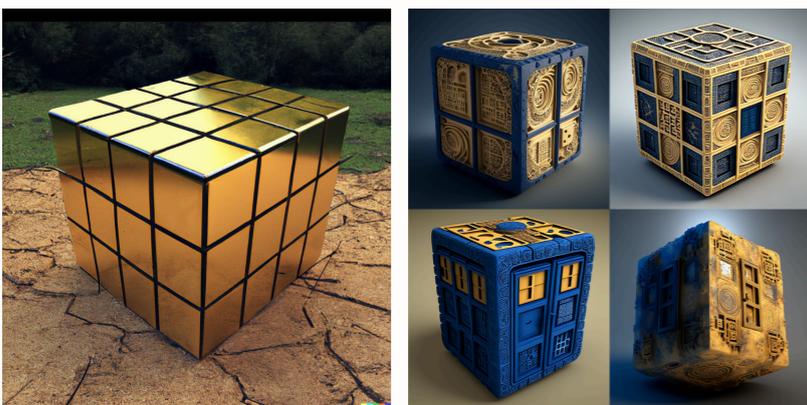


# GANs et Text to image



An Astronaut riding a horse in a photorealistic style →

↓ Giant gold Rubik's cube on the ground on earth...



a Menger sponge Tardis ↑

Dall-E 2 et 3, this X does not exist, This person does not exist,



Sora (Open AI, 2024) (observer la seconde 16) Vidéo



[Movie Gen \(Meta, 2024\) / blog post](#)

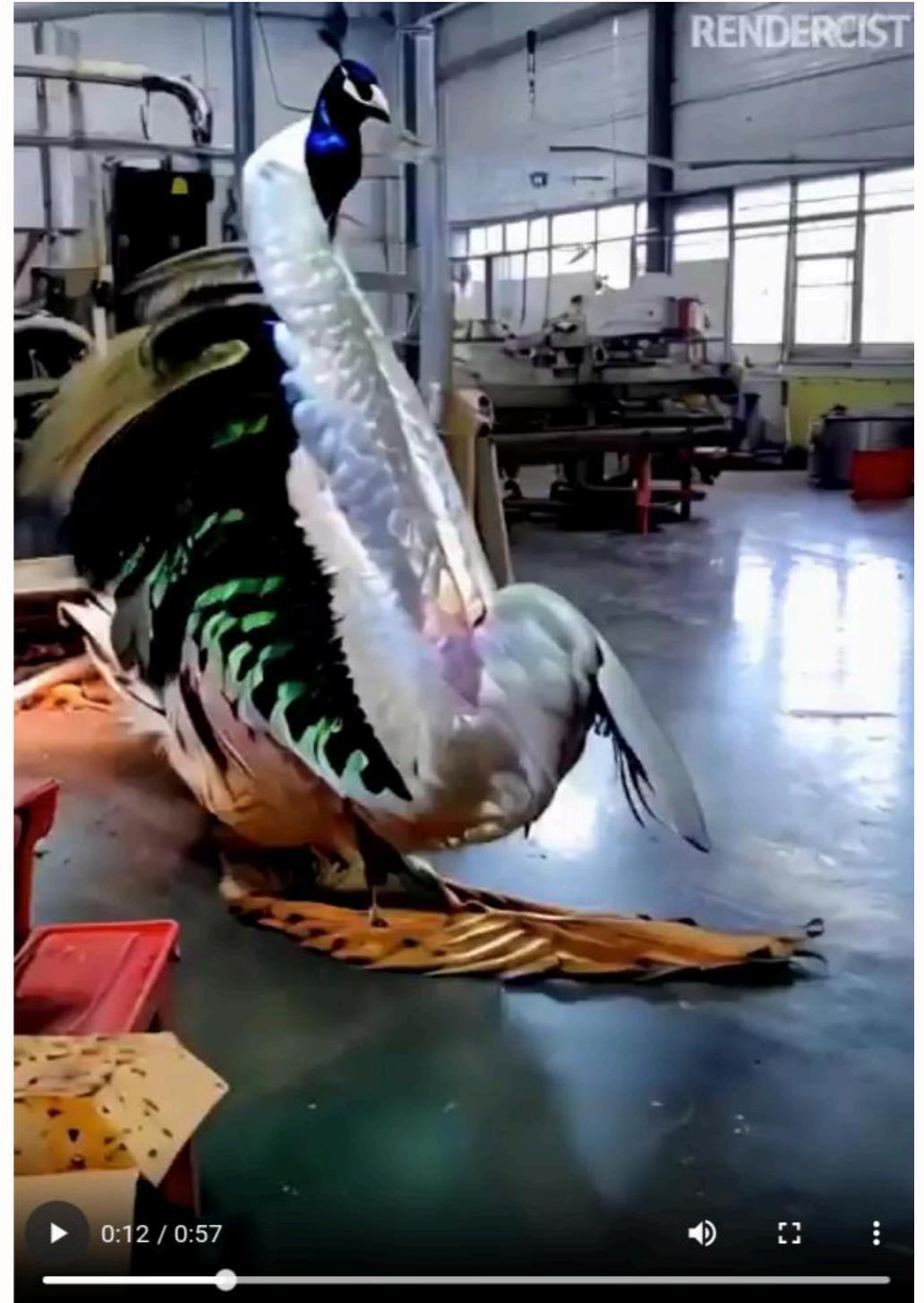


# Démonstration de Deep-fakes (The Dor Brothers)

## Vidéo



# IA sous LSD (post de [Justine Moore ?](#))



# Génération de texte

# Principe des LLMs

- réseaux de neurones
- opère sur des nombres
- formalisable sous forme de calcul matriciel
- architecture *Transformer* (2017) : **Attention is all you need**



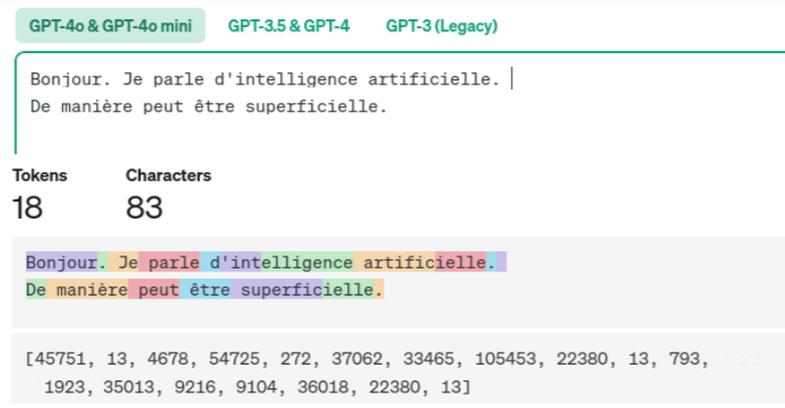
- modèle de fondation (GPT, Llama,...) : **complète un texte en prévoyant le mot suivant**
- apprentissage par renforcement pour avoir un chatbot (ChatGPT, Gemini...) : produit une discussion complète
- modèles « avec réflexion » (ChatGPT o1, DeepSeek r1)

The screenshot shows a text generation interface. The main text area contains a story in French: "C'est l'histoire d'un élève qui voulait faire un programme ne utilisant une IA. Il s'est connecté sur le service et alors qu'il était en train de télécharger le logiciel nécessaire, il a reçu un message lui disant que le service était en maintenance et qu'il ne pourrait pas accéder au programme. Il était déçu et décida de trouver une autre manière de réaliser son projet. Il se mit alors à chercher et trouva un autre service qui lui permettait d'utiliser une IA. Il se connecta donc et commença à travailler sur son projet. Il mit tout son cœur et toute son énergie dans ce projet et après plusieurs heures de travail, il parvint à terminer son programme et à le mettre en ligne. Il fut très fier de son travail et fut très heureux de voir que son progra". Below the text, a probability distribution for the next token is shown: cr = 35.57%, le = 25.79%, ob = 2.51%, ré = 13.49%, termin = 7.36%. A tooltip for "termin" shows "Total: -2.61 logprob on 1 tokens (84.72% probability covered in top 5 logits)". On the right, there is a control panel with a dropdown menu set to "text-davinci-003", and sliders for "Maximum Length" (256), "Temperature" (0.5), "Top P" (1), "Frequency Penalty" (0), and "Presence Penalty" (0). There are also checkboxes for "Show Probabilities" and "Highlight Models", and a "Reset Parameters to default" button.

# Que fait le LLM ?

## Entrée : Un texte

### 1. Découpage en mots/token (*tokenisation*) → Tokenizer GPT



### 2. Plongement (*embedding*)

### 3. Transformer (*decoder*) (Étapes répétées 96 fois dans GPT 3)

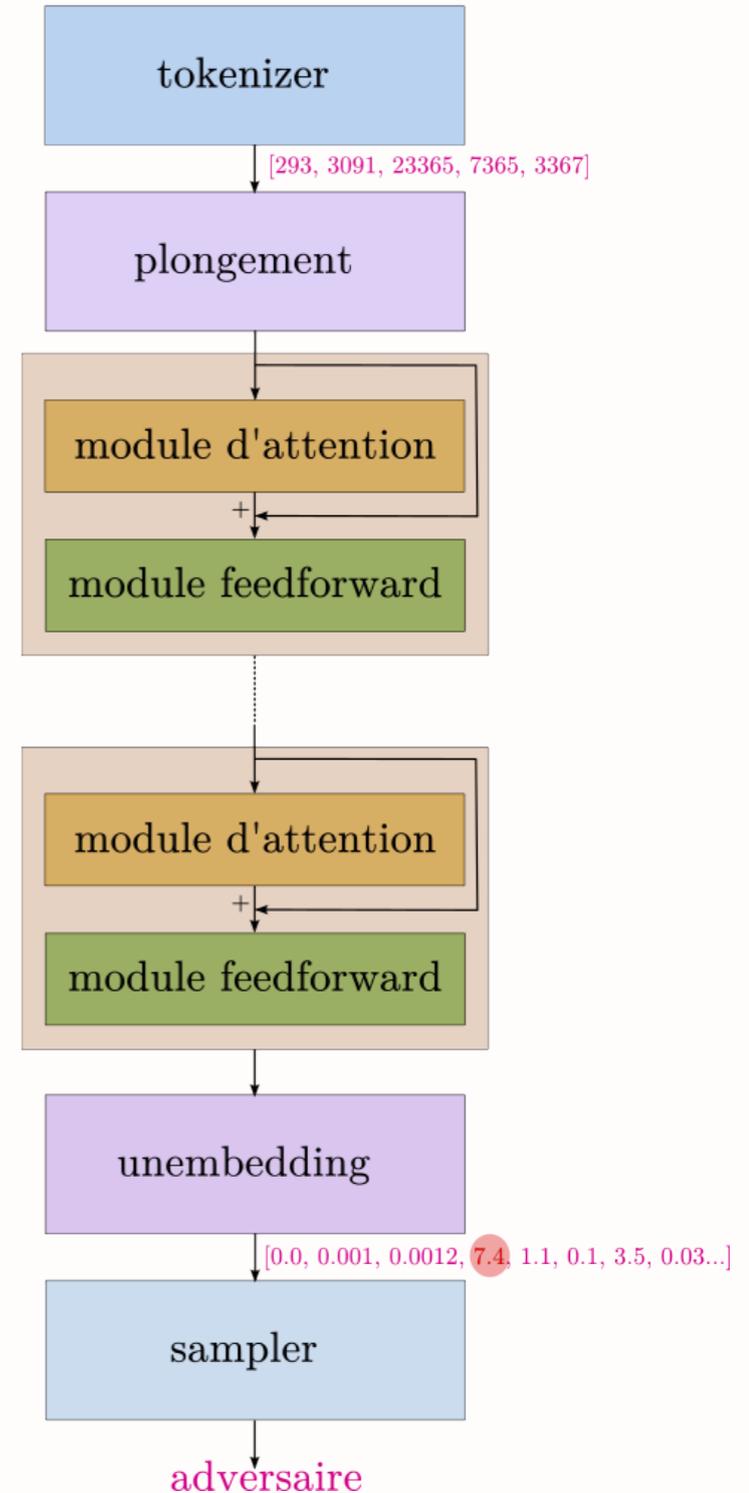
### 4. Unembedding

### 5. Sampler

## Sortie : un mot à ajouter au texte

Cas GPT 3 : plongement (12 288) / nb tokens (50 257)

le boxeur bat son



## Comment représenter des tokens/mots/idées par des nombres ?

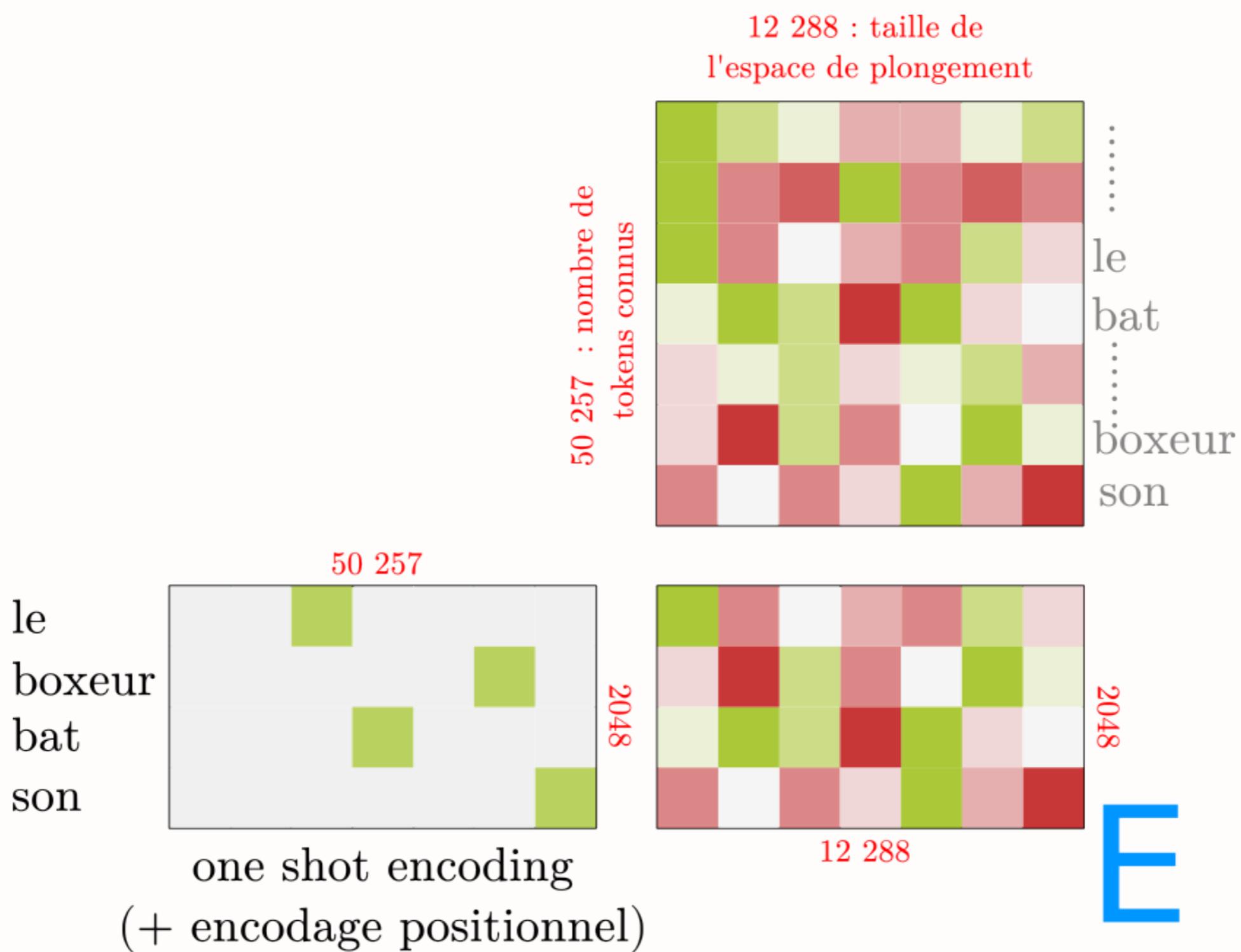
- **Plongement dans un espace de grande dimension**
- Un token → un vecteur (dim 12288 pour GPT3)
- Plongement tel que : des concepts similaires sont proches dans l'espace de plongement
- **Calcul vectoriel avec signification sémantique**

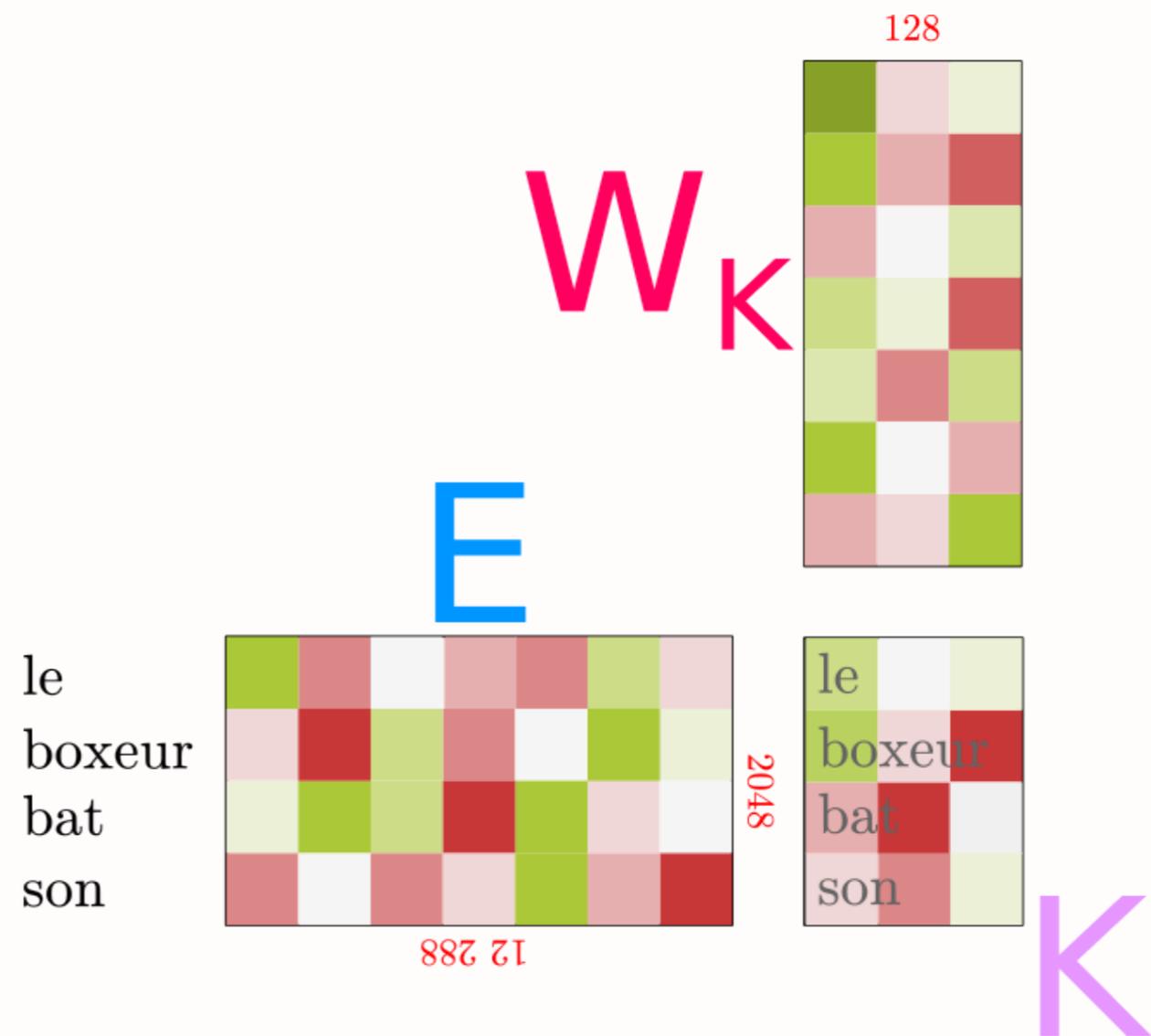
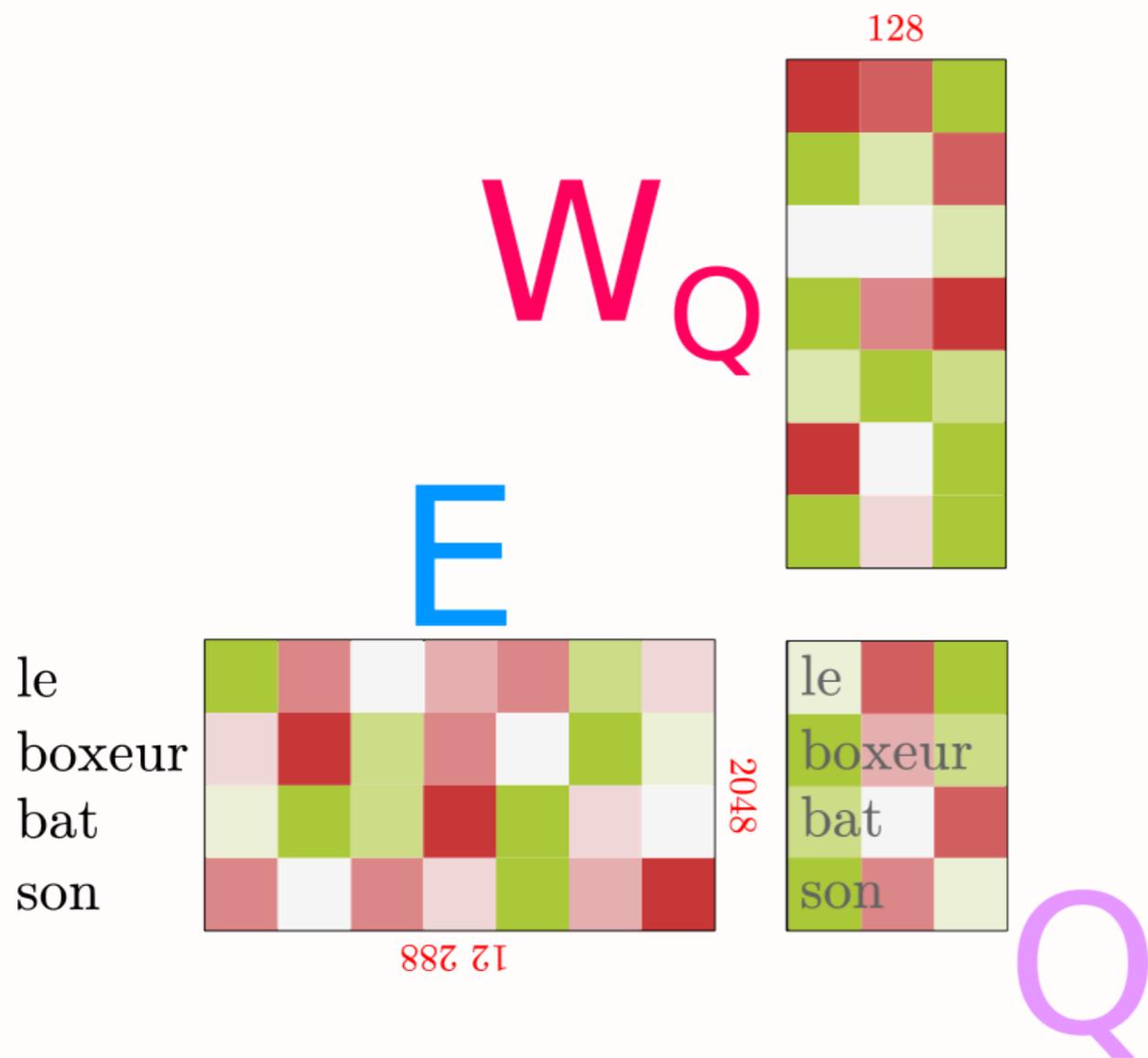
```
>>> model2 = gensim.downloader.load('glove-wiki-gigaword-300')
>>> model2['father']
array([-1.3874e-01, -8.3454e-02,  3.0141e-01, -1.2189e-01, -1.1167e-01,
       ...,
       7.2142e-02,  4.2937e-01, -1.3576e-01, -5.2199e-01,  6.7699e-01])
```

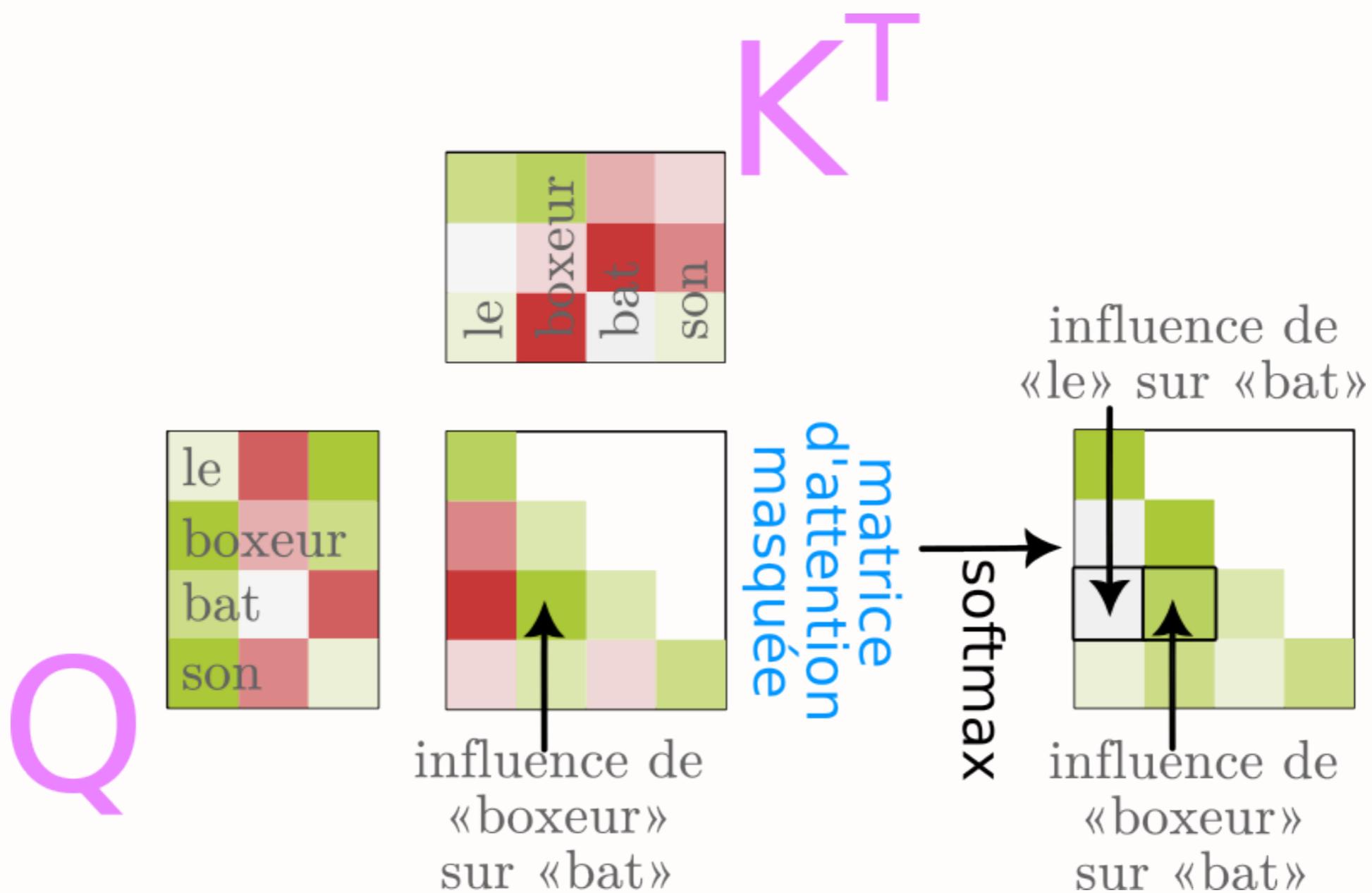
```
>>> v = model2['man'] - model2['woman'] + model2['mother']
array([-1.27260000e-01,  1.11300983e-02, -1.62580013e-01, -1.90102994e-01,
       ...,
       2.45849997e-01, -2.34270006e-01, -4.49473977e-01,  6.22990072e-01])
>>> model2.most_similar(v, topn=1)
[('father', 0.7911101579666138)]
```

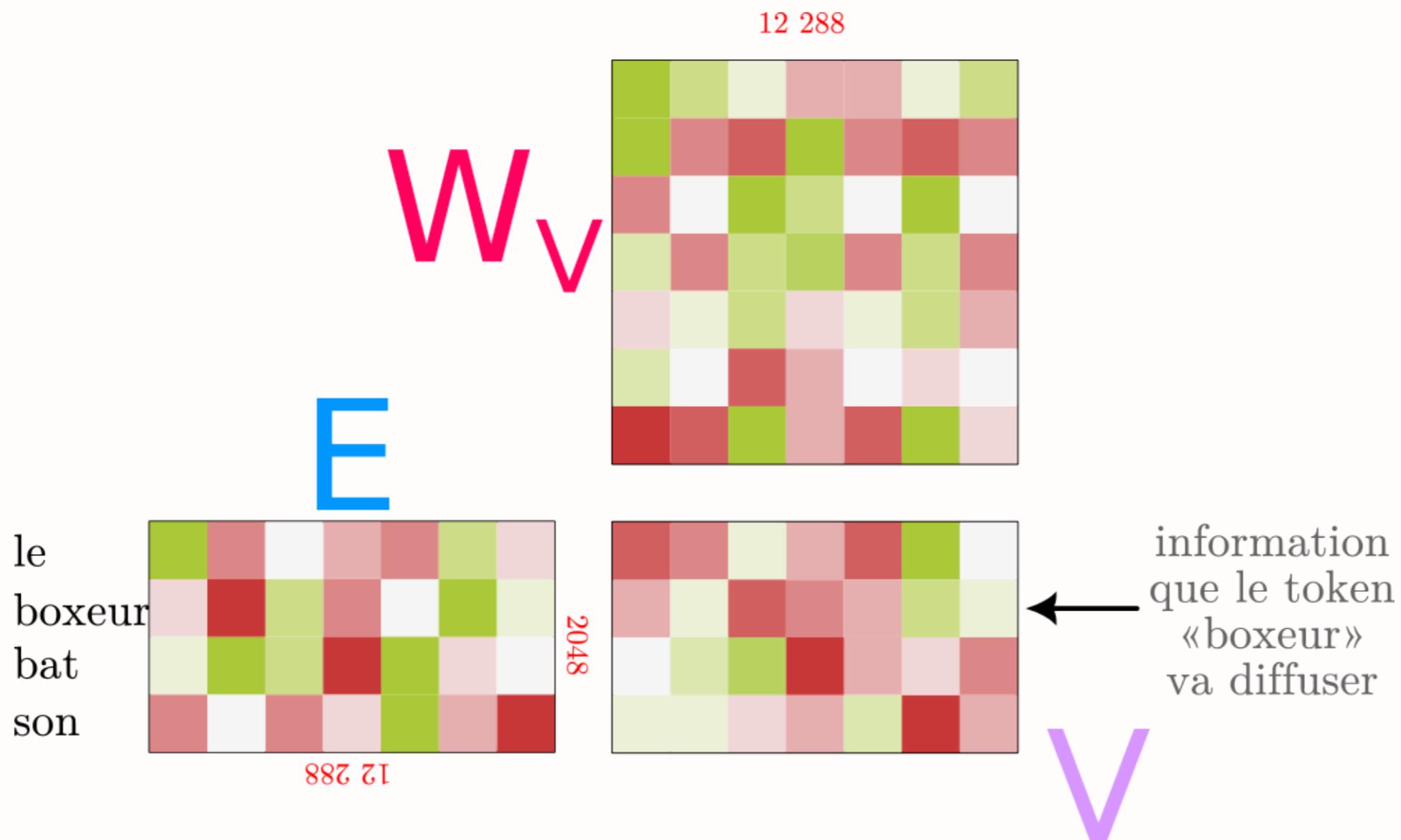
```
>>> model2.most_similar(model2['man'] - model2['woman'] + model2['girl'], topn=1)
[('boy', 0.814935028553009)]

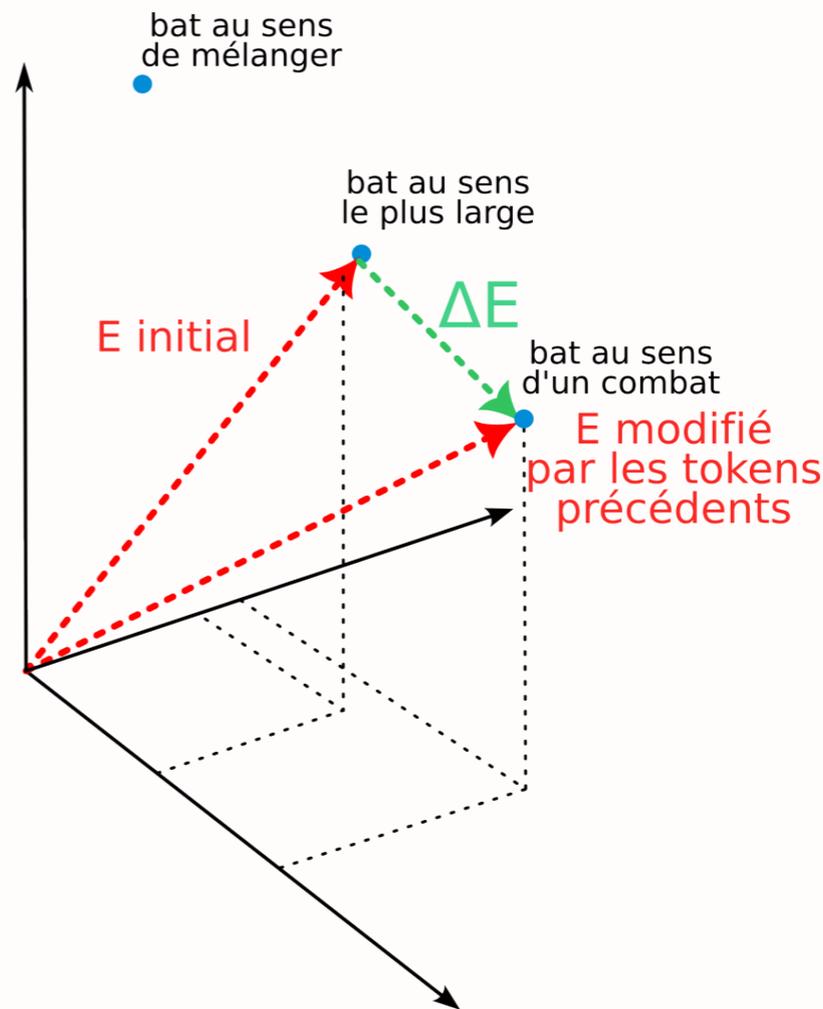
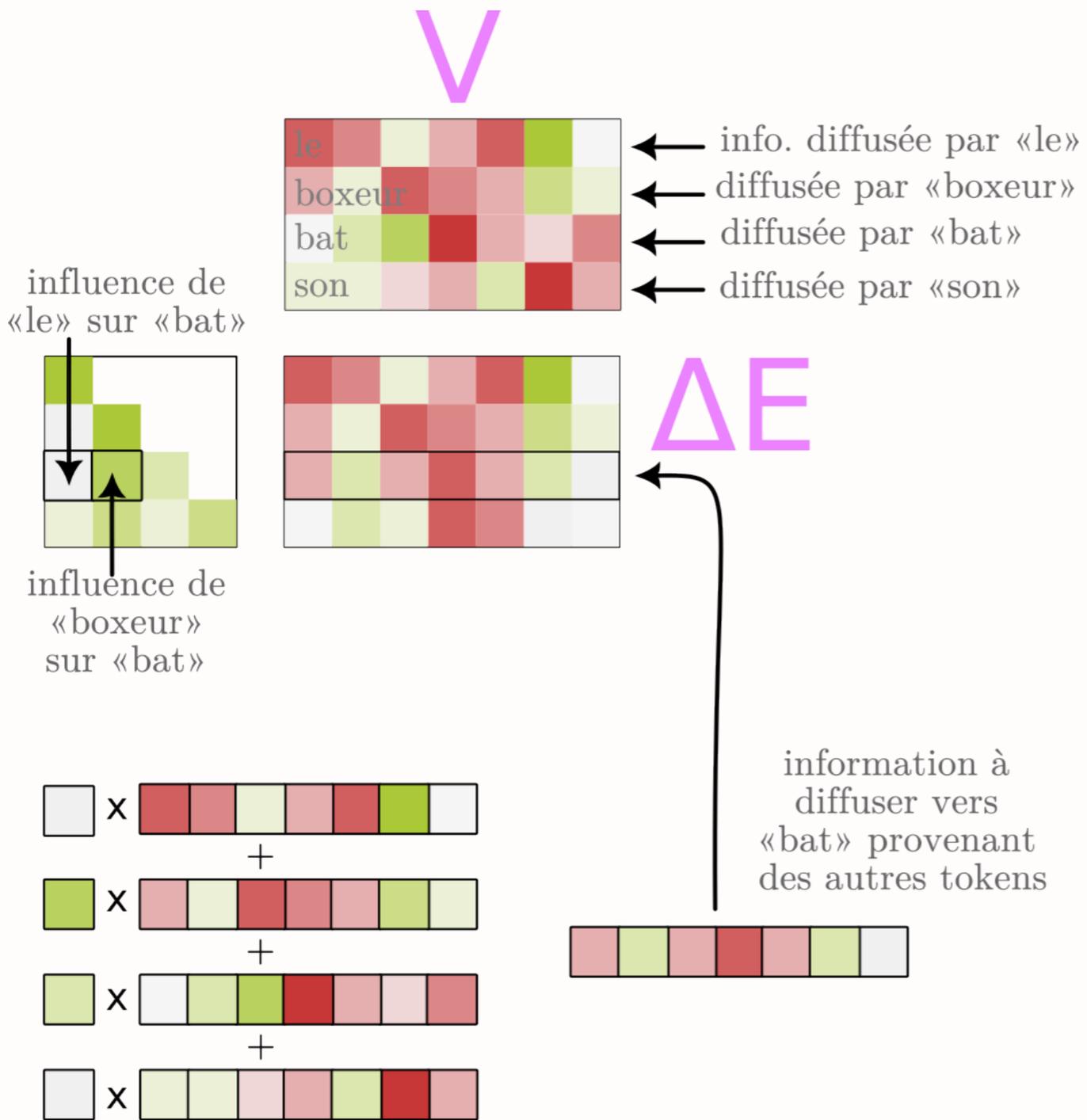
>>> model2.most_similar(model2['hitler'] - model2['germany'] + model2['italy'], topn=1)
[('mussolini', 0.6974261999130249)]
```



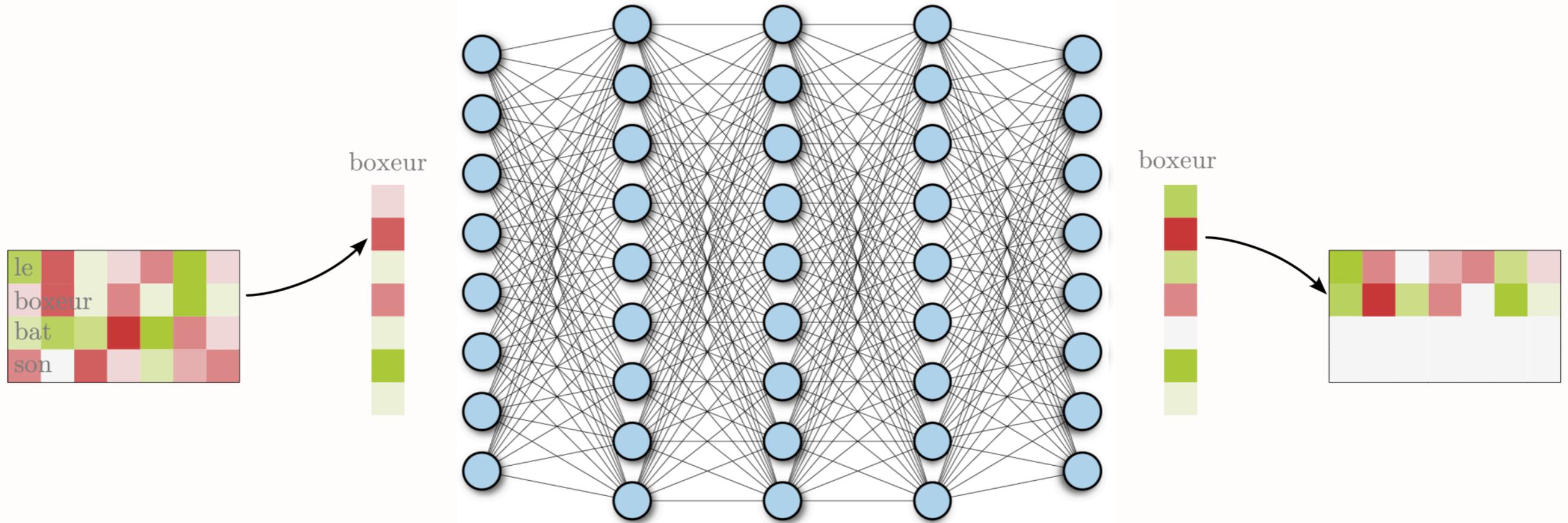




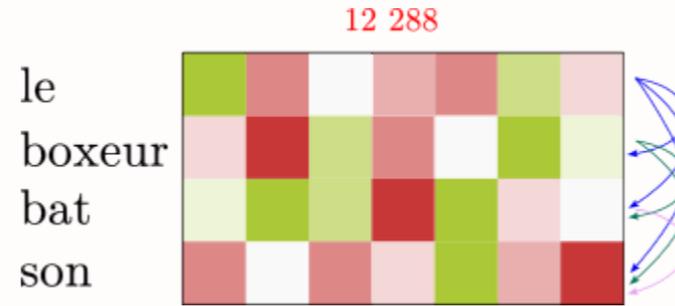




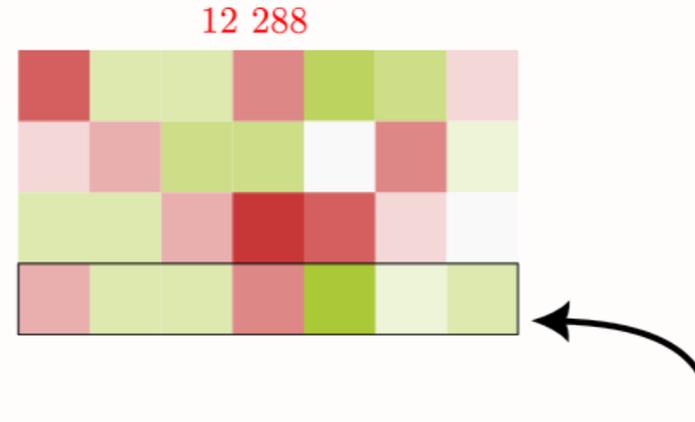
Le réseau Feed-forward est une «simple» transformation de chaque token, indépendamment les un des autres.



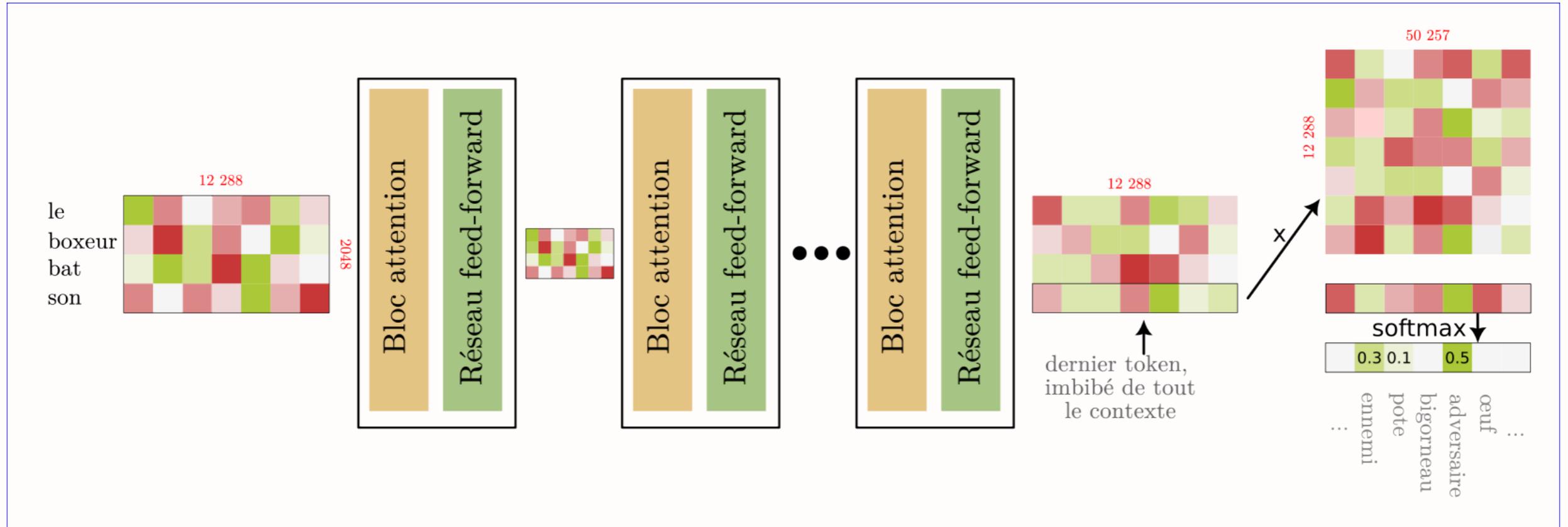
Au fil des couches, **enrichissement** des tokens par l'**attention** et le **feed-forward**.

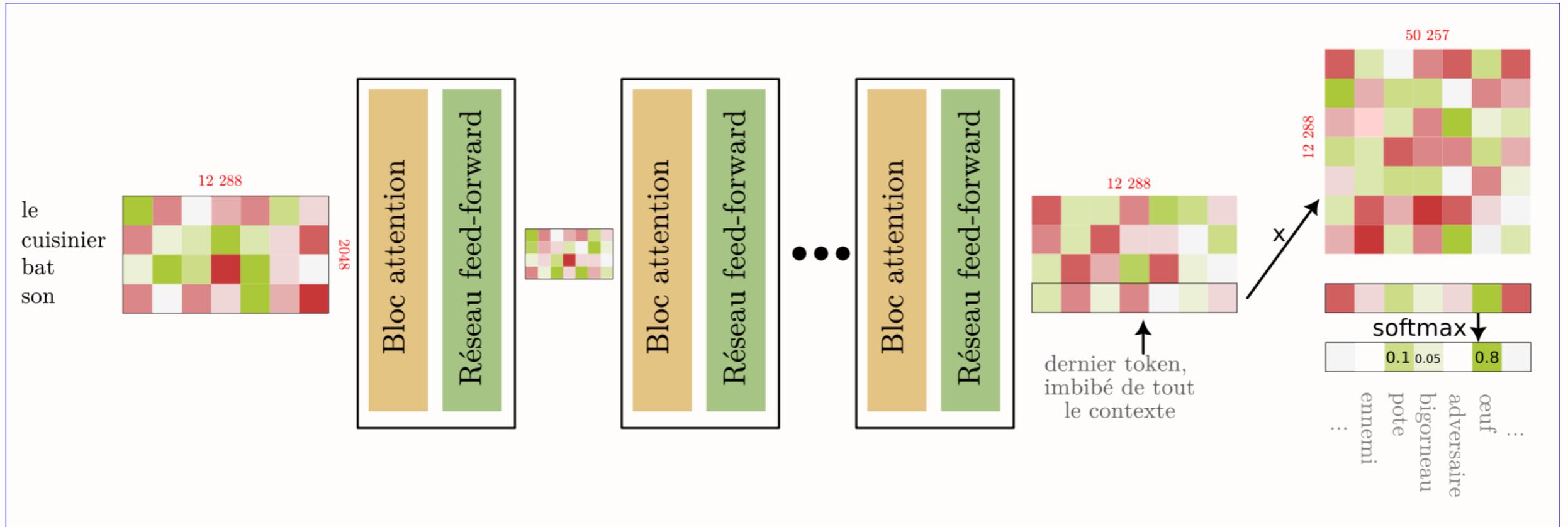


Après plusieurs couches, les tokens ne correspondent plus aux tokens de départ. Mais on espère que **chaque vecteur contient toute l'information nécessaire à la prédiction du token qui le suit**.



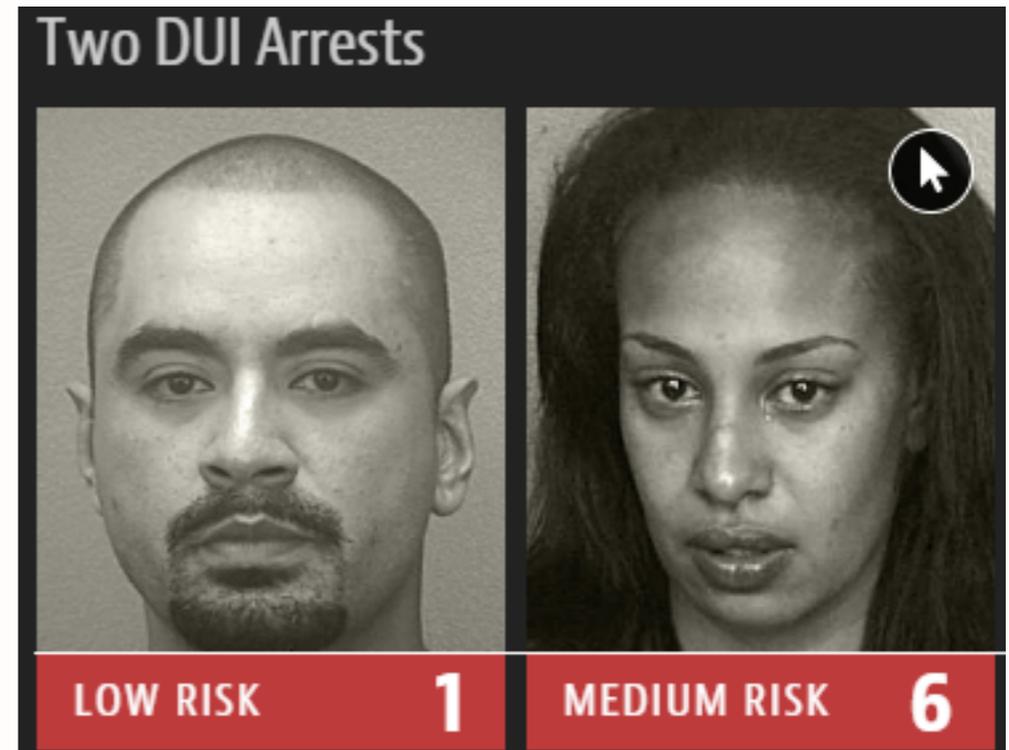
idée d'un boxeur, de combat, de victoire  
(toute l'information qui permet  
de prévoir le mot suivant)





# Biais

- le corpus d'apprentissage utilisé (images, texte) est humain
- ses défauts seront reproduits (amplifiés) par la machine



## Quelques exemples

- arrestation de Nijeer Parks à tort (2019)
- 100% de réussite pour la reconnaissances d'hommes blancs, mais seulement 70 % pour des femmes noires
- génération d'images et de texte / traduction qui reflète les biais de notre société

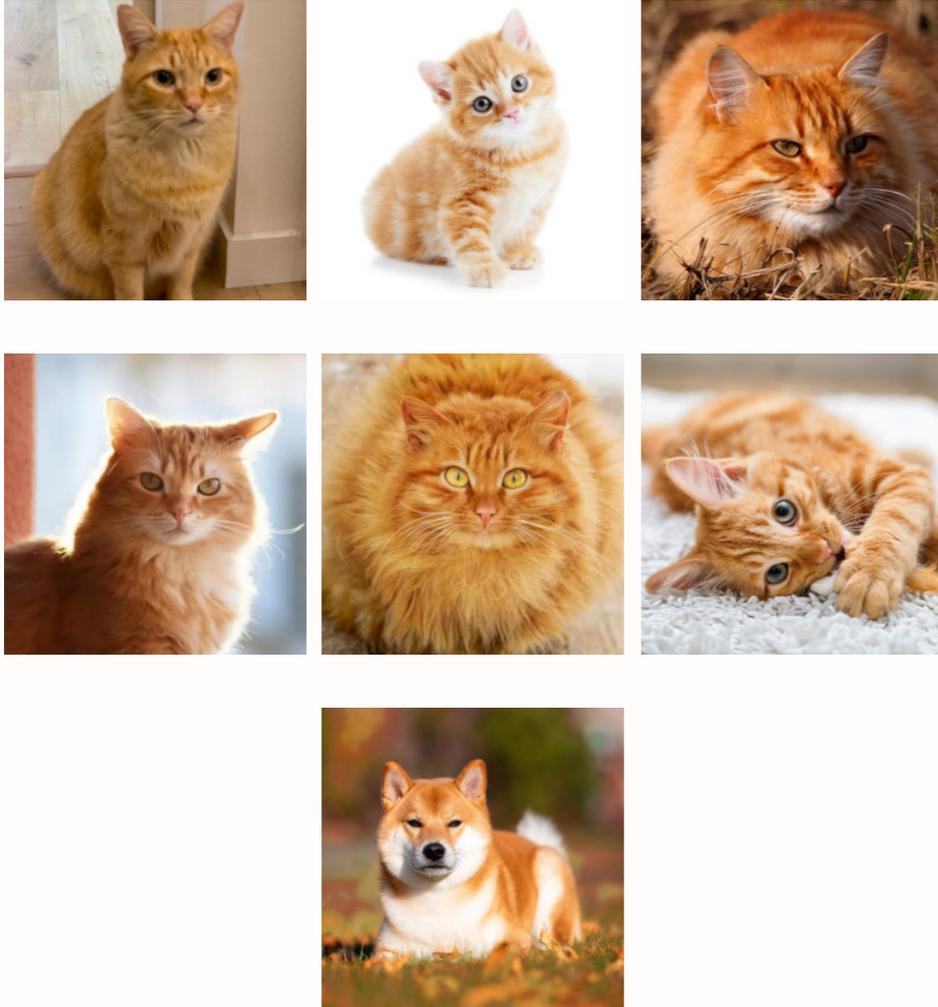
The screenshot shows the Google Translate interface with the source language set to 'Anglais' and the target language to 'Français'. A list of professions is entered in the source box, and the translated text is shown in the target box. The translations are as follows:

Anglais	Français
a nurse	une infirmière
a doctor	un docteur
a scientist	un scientifique
a beutician	une esthéticienne
a novel writer	un romancier
a secretary	une secrétaire

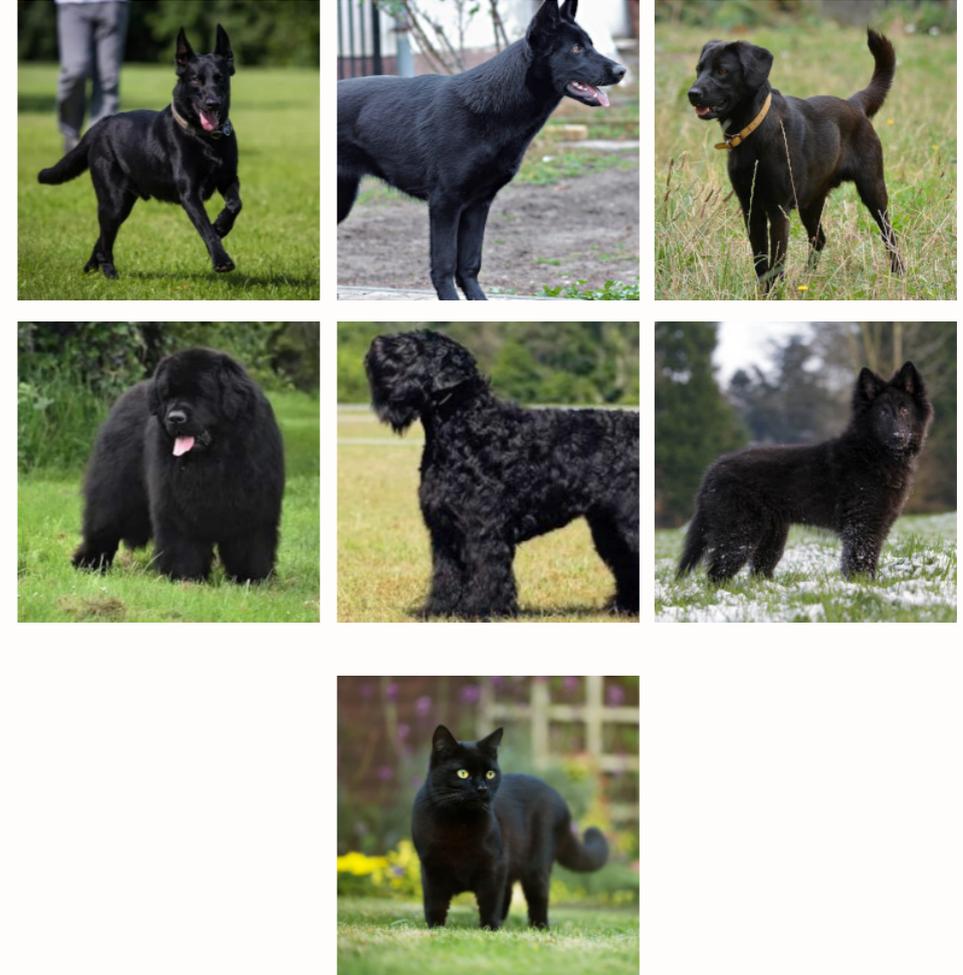
At the bottom of the interface, there are links for 'Ouvrir dans Google Traduction' and 'Commentaires'.

# D'où viennent les biais : exemple jouet

## Chats

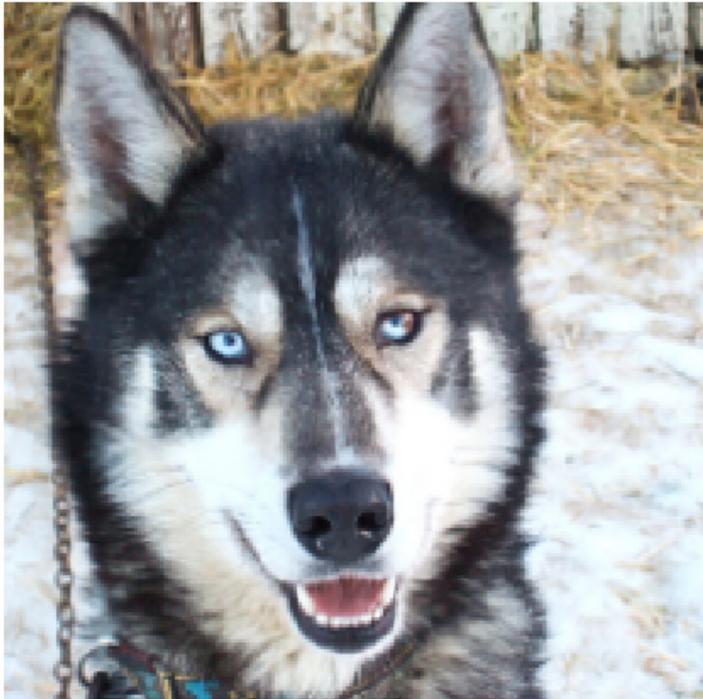


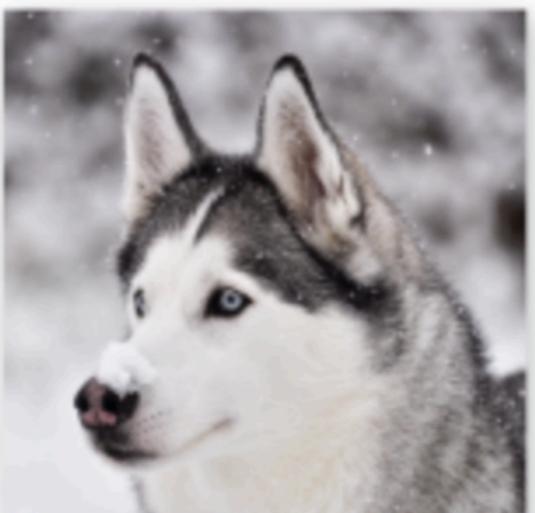
## Chiens



## Biais : autres exemples ( $\approx$ réels)

- battes de base-ball et bras
- un réseau a été entraîné à reconnaître les images contenant un animal des autres images.
- reconnaître **loups et huskies**



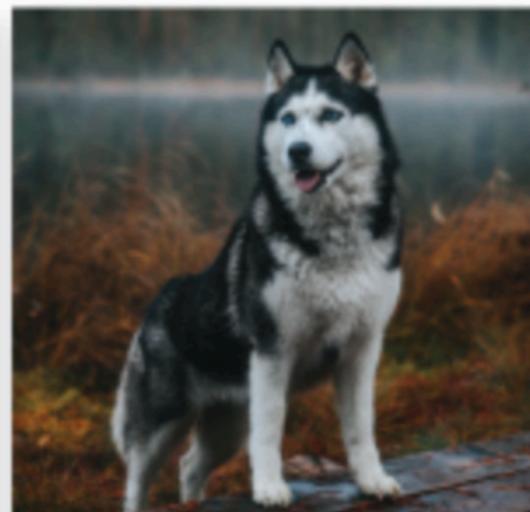




Predicted: **Wolf**  
True: **Wolf**



Predicted: **Husky**  
True: **Husky**



Predicted: **Husky**  
True: **Husky**



Predicted: **Wolf**  
True: **Wolf**



Predicted: **Wolf**  
True: **Wolf**



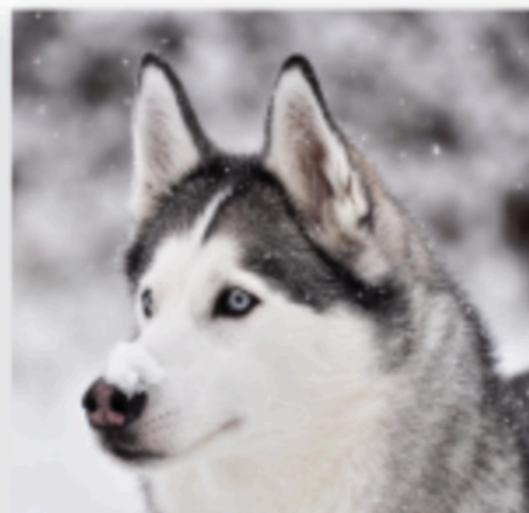
Predicted: **Wolf**  
True: **Wolf**



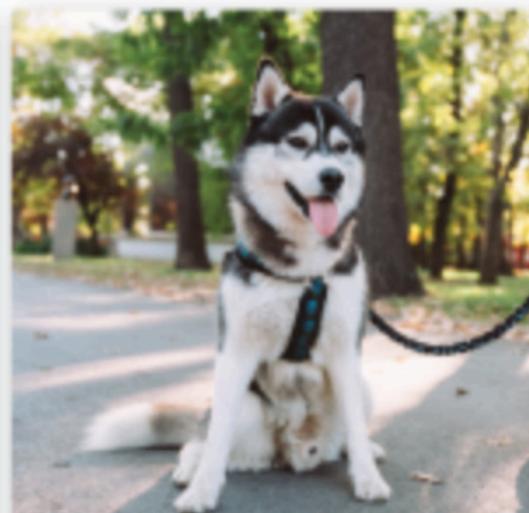
Predicted: **Husky**  
True: **Wolf**



Predicted: **Wolf**  
True: **Wolf**



Predicted: **Wolf**  
True: **Husky**

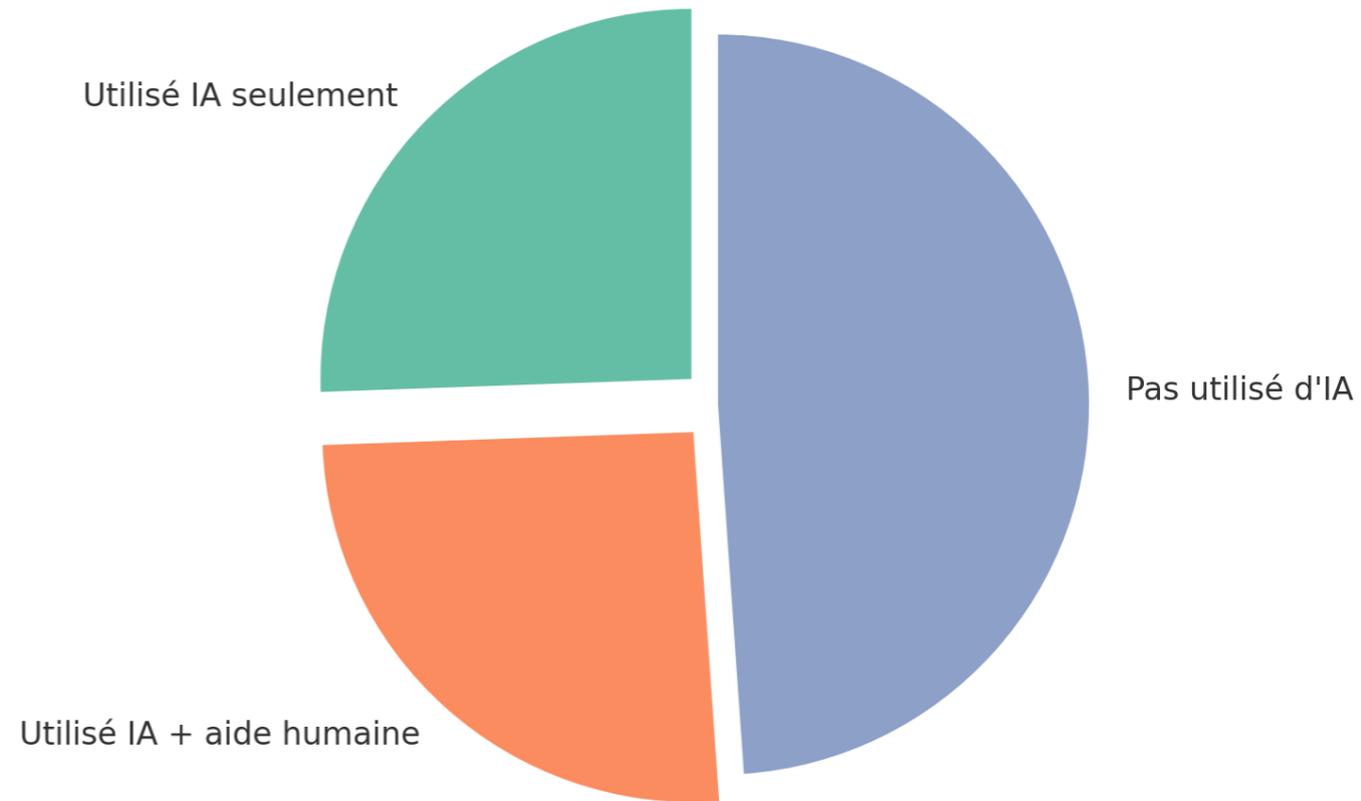


Predicted: **Husky**  
True: **Husky**

# Apprentissage de la programmation

- Comment justifier l'apprentissage de la prog auprès de futurs ingénieurs scientifiques non-informaticiens ?
- Résultat d'un sondage :
  - Contexte : exercices à faire chez soi (entre les TDs)
  - 90 pers. interrogées (BAC + 3, 1er année école d'ingé, après une prépa)
  - La moitié des étudiants ayant reçu l'aide d'une IA pensent ne toujours pas savoir faire l'exercice sans aide.
  - On parle d'exercices de moins de 10 lignes.

Répartition des réponses sur l'utilisation d'une IA



## À quoi pourraient être utilisés les LLMs ?

Hors problème d'évaluation :

1. **débloquer une situation** : les élèves doivent produire un rapport circonstancié sur ce qui bloquait et pourquoi ça ne bloquera plus
2. **progresser** :
  - vérifier son propre code (déjà fonctionnel ou buggé)
  - améliorer le côté idiomatique, la virtuosité
  -  faisable et utile dès maintenant