

*Groupe Informatique*

*IREM de Lille*

*Asli Grimaud*



# IA et Apprentissage Automatique

*Avril 2026*

`asli.grimaud@ac-lille.fr`

## Apprentissage supervisé

- Algorithme des  $k$ -plus proches voisins
- Algorithme ID3

Wikipédia (en) : A large language model (LLM) is a language model trained with self-supervised machine learning on a vast amount of text, designed for natural language processing tasks, especially language generation.

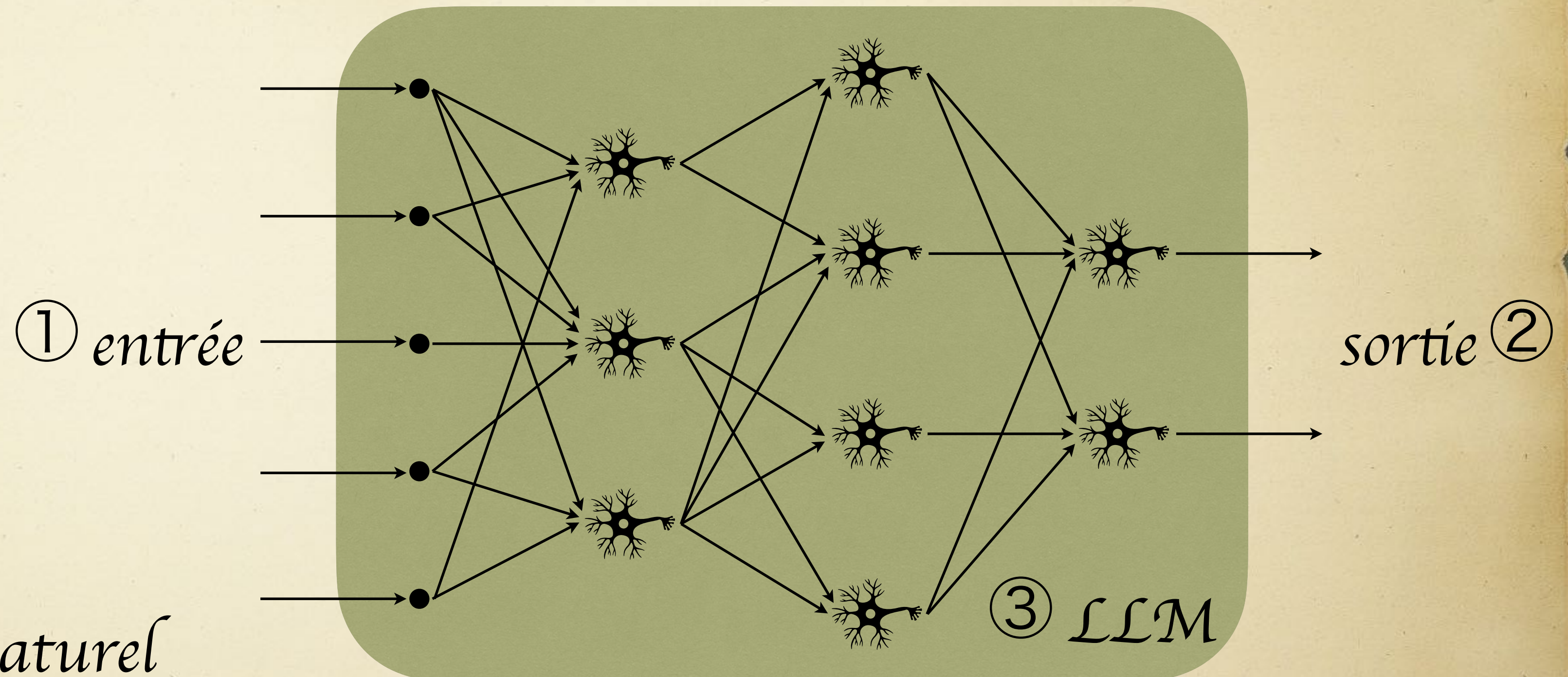
## LLM

- Réseaux de neurones
- Spécialisations

## Apprentissage non supervisé

- Algorithme des  $k$ -moyennes
- Classification hiérarchique ascendante

# Intelligence artificielle - Fonctionnement d'un chatbot LLM

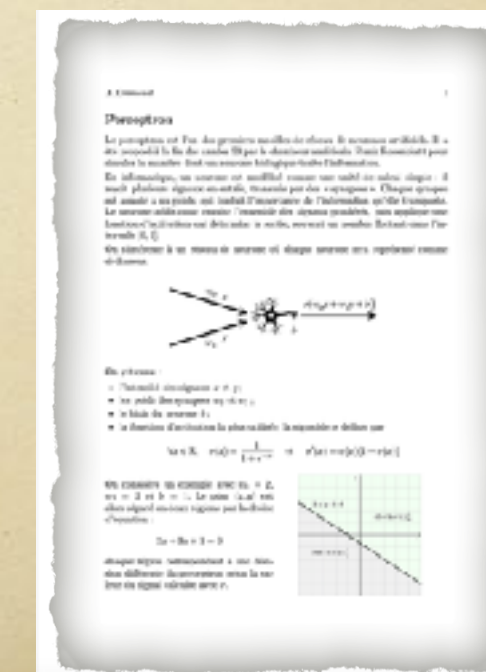


① Traduction d'un langage naturel

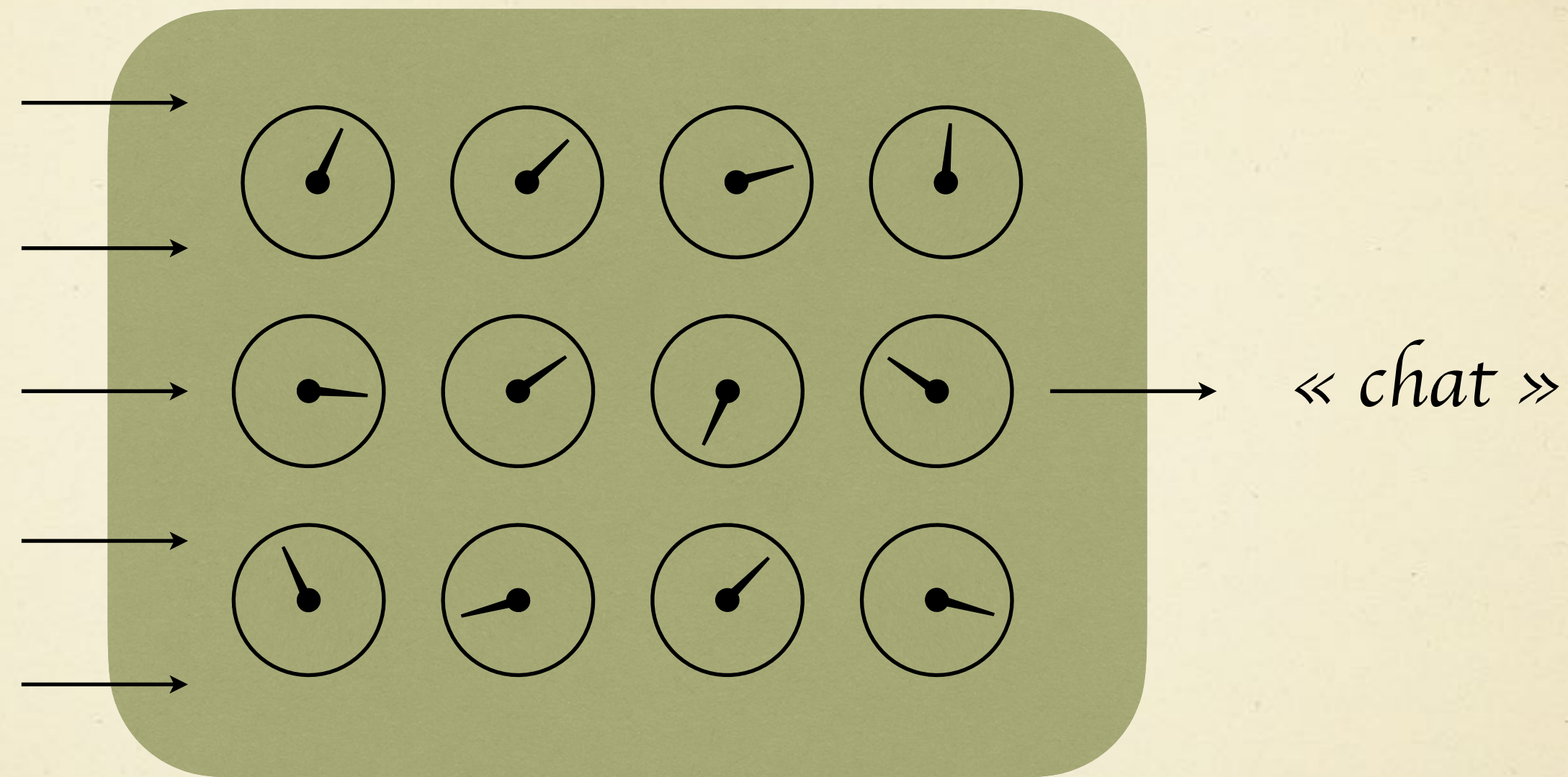
② Modèle de fondation et spécialisation vers un chatbot

③ Réseaux de neurones

④ Implémentation en Python d'un perceptron affine (ou-exclusif)

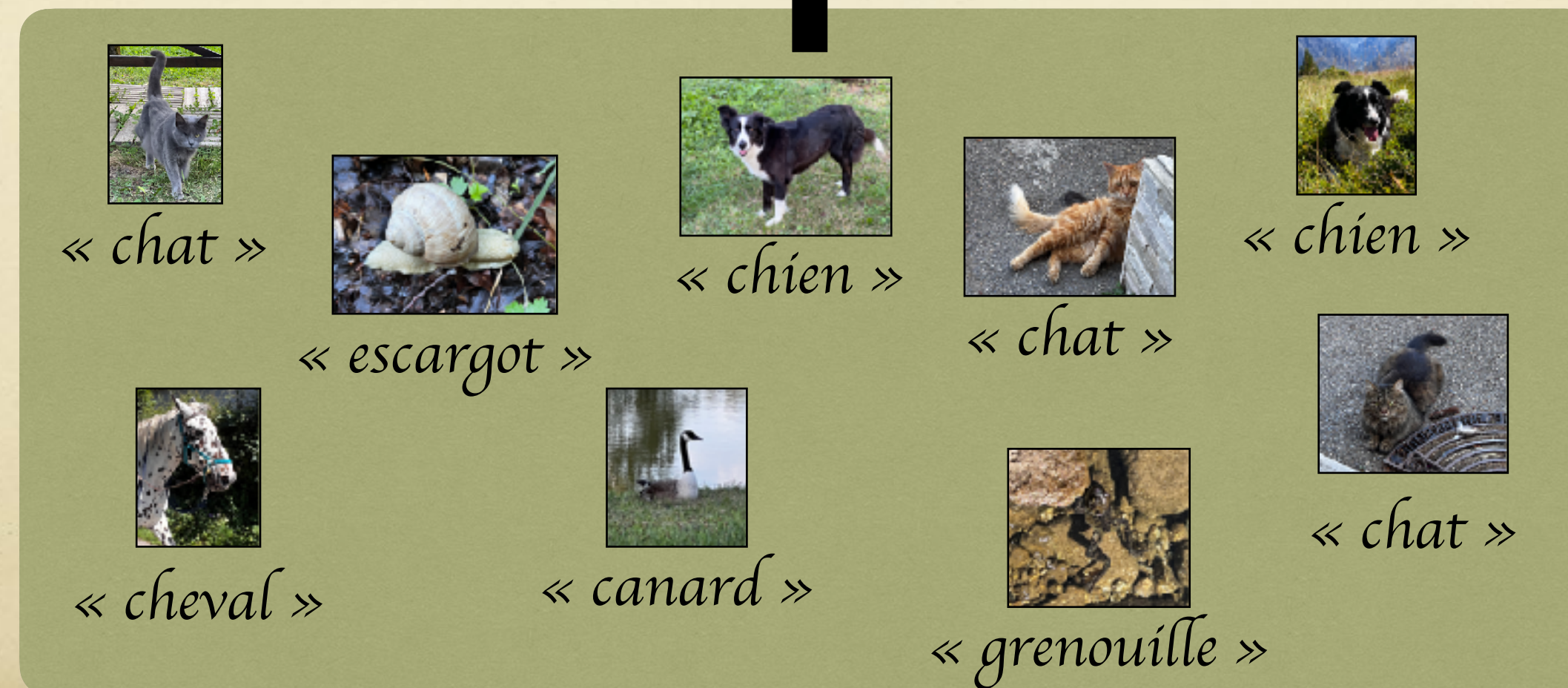
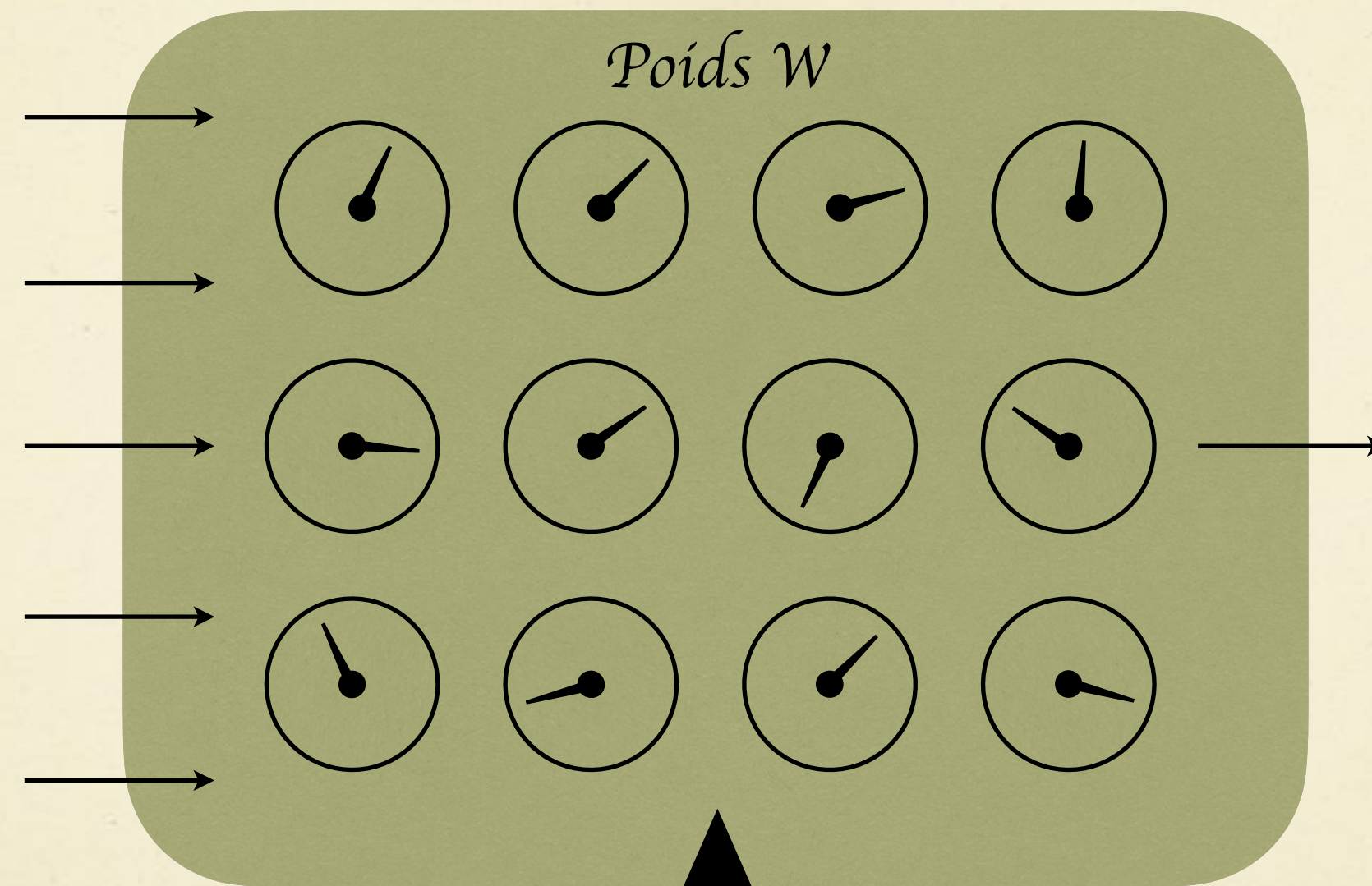


# Apprentissage supervisé : Reconnaissance d'images



# Apprentissage supervisé : Reconnaissance d'images

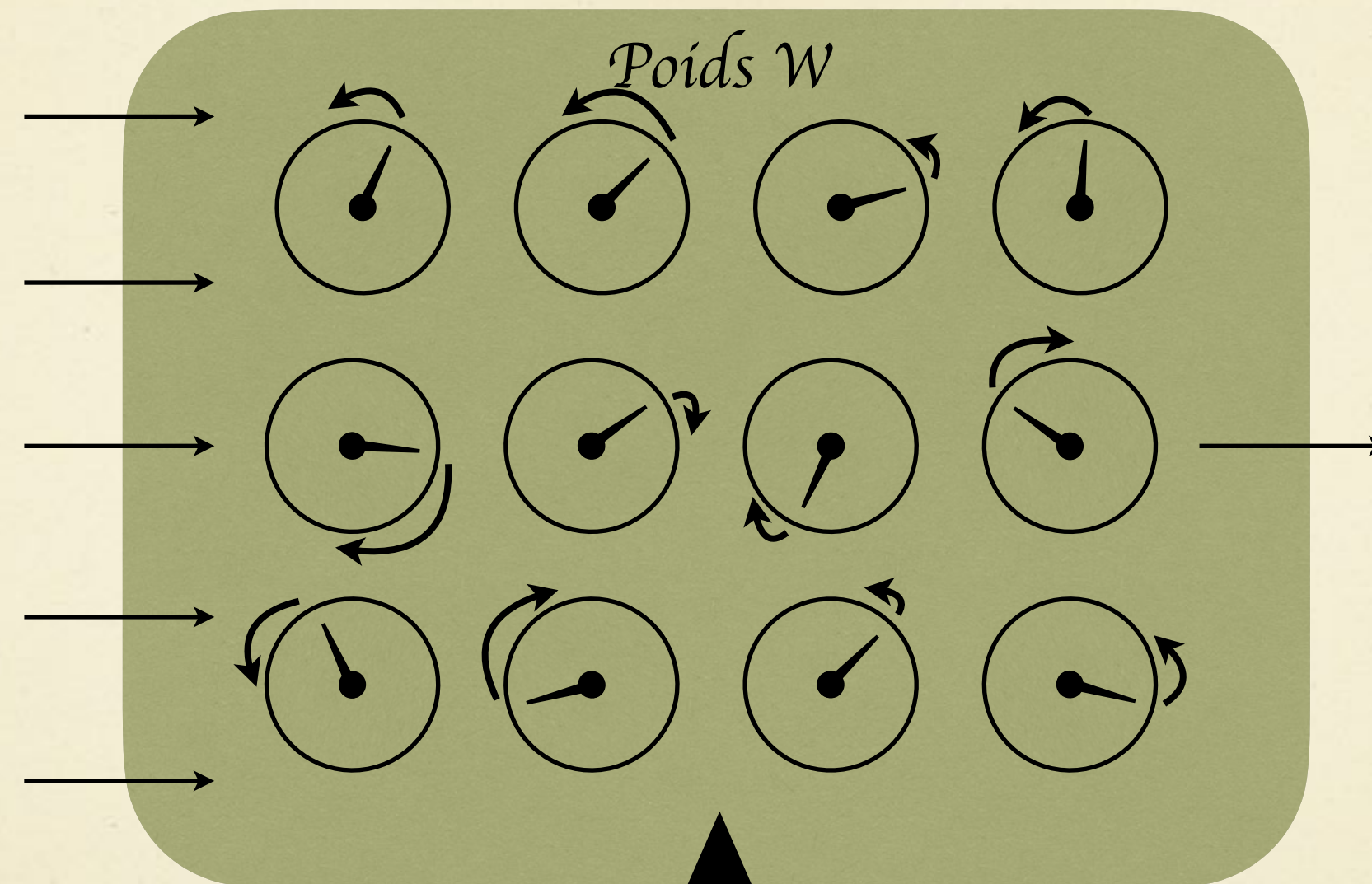
Phase d'entraînement



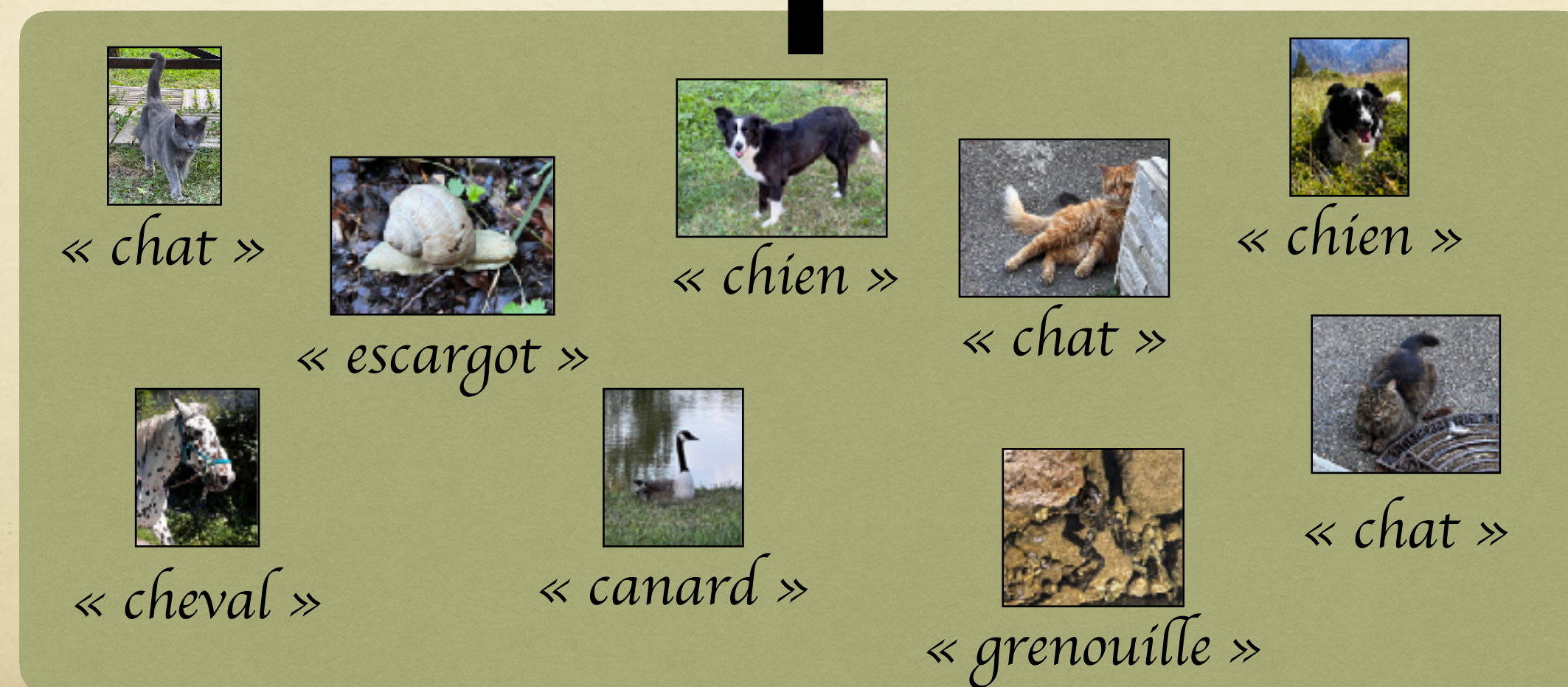
Ensemble  d'apprentissage

# Apprentissage supervisé : Reconnaissance d'images

Phase d'entraînement



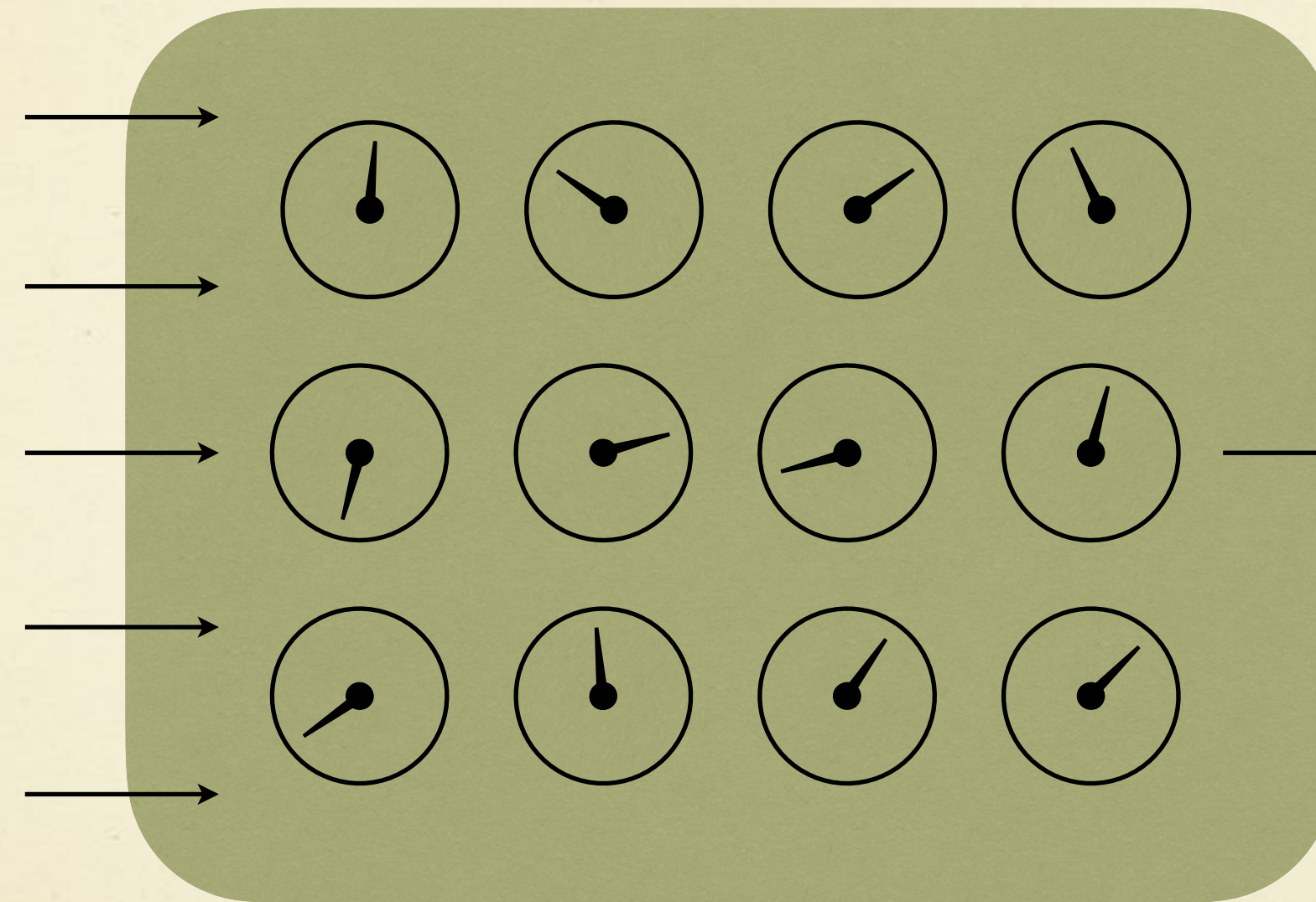
Objectif : ajuster les poids pour coller aux exemples de l'ensemble d'apprentissage



Ensemble  d'apprentissage

# Apprentissage supervisé : Reconnaissance d'images

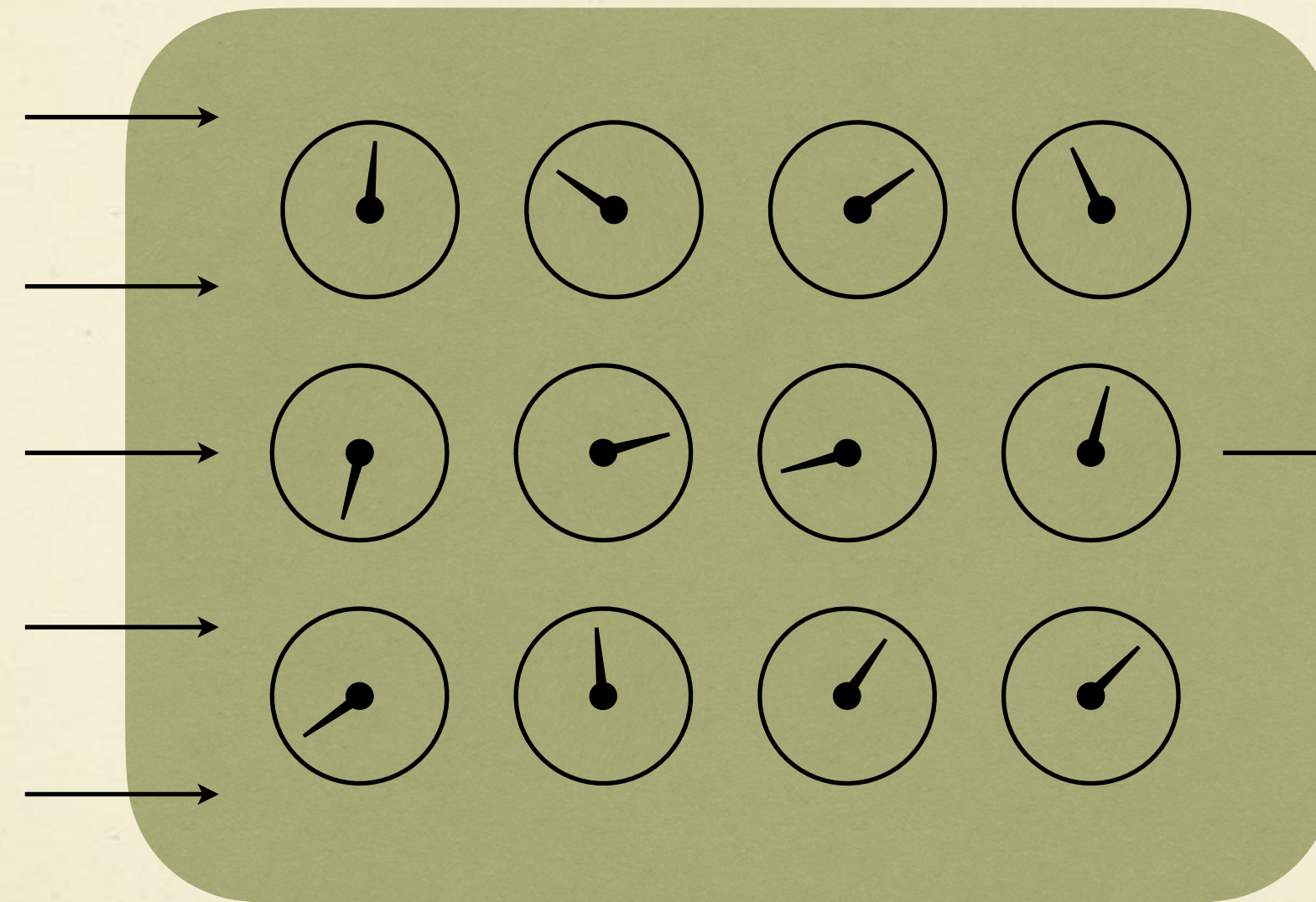
Phase d'utilisation



« chien »

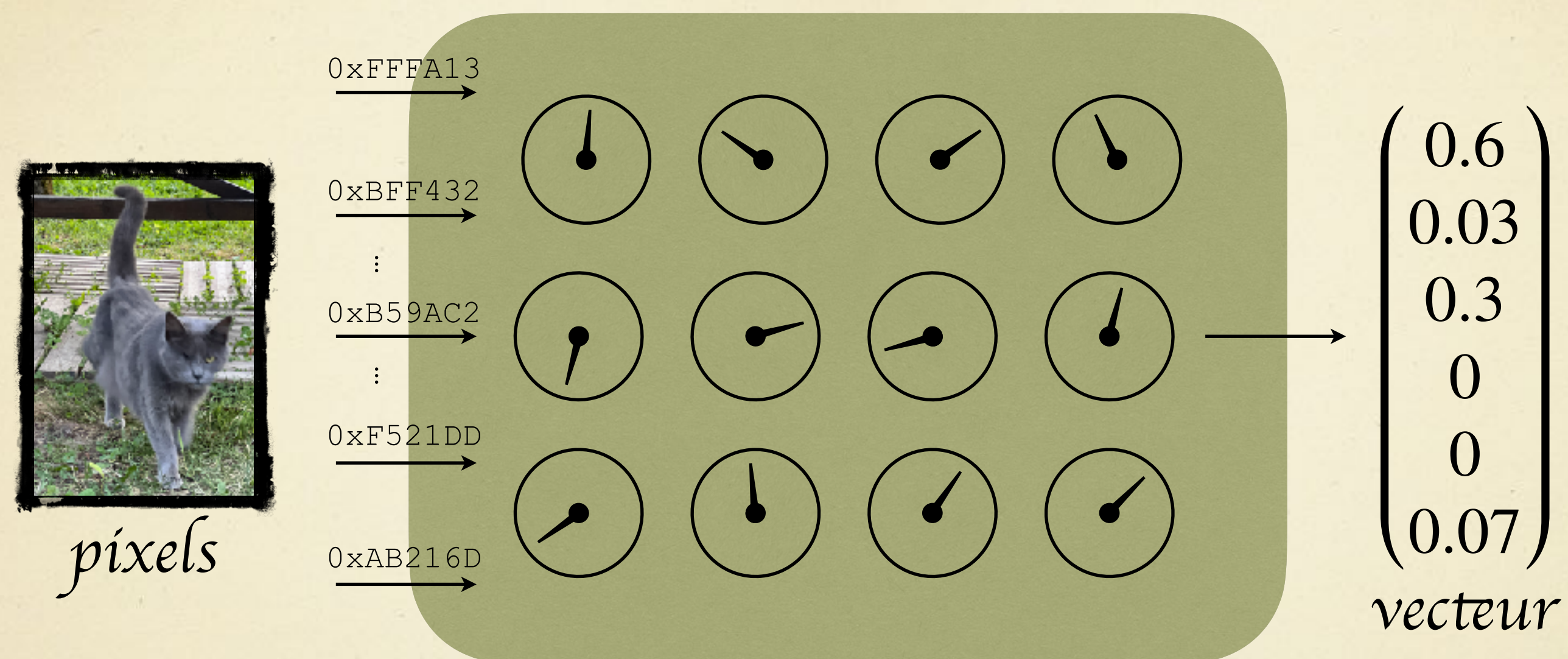
# Apprentissage supervisé : Reconnaissance d'images

Phase d'utilisation



« escargot »

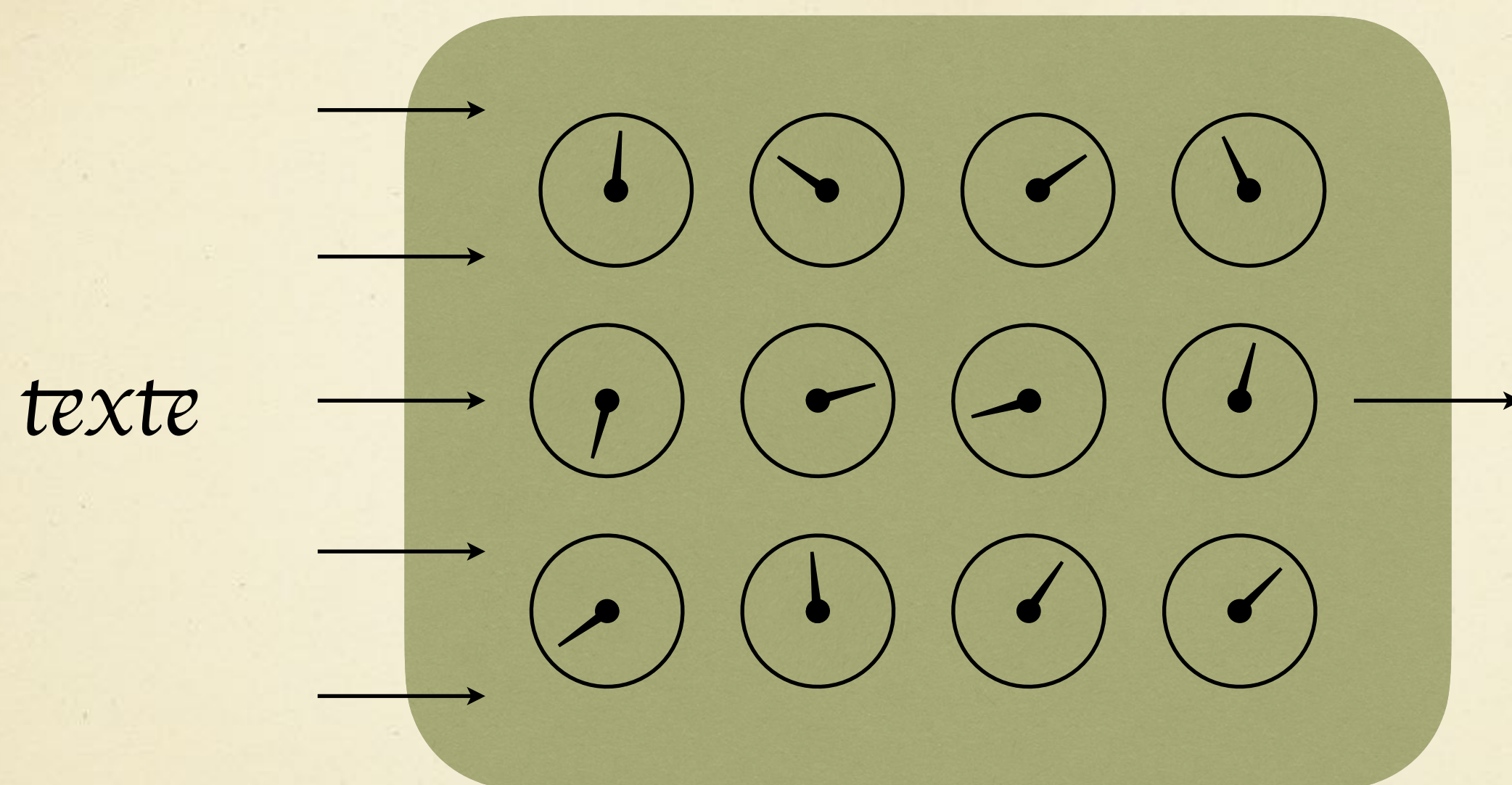
# Apprentissage supervisé : Reconnaissance d'images



« chat » : choix de classe au hasard parmi les plus probables

①

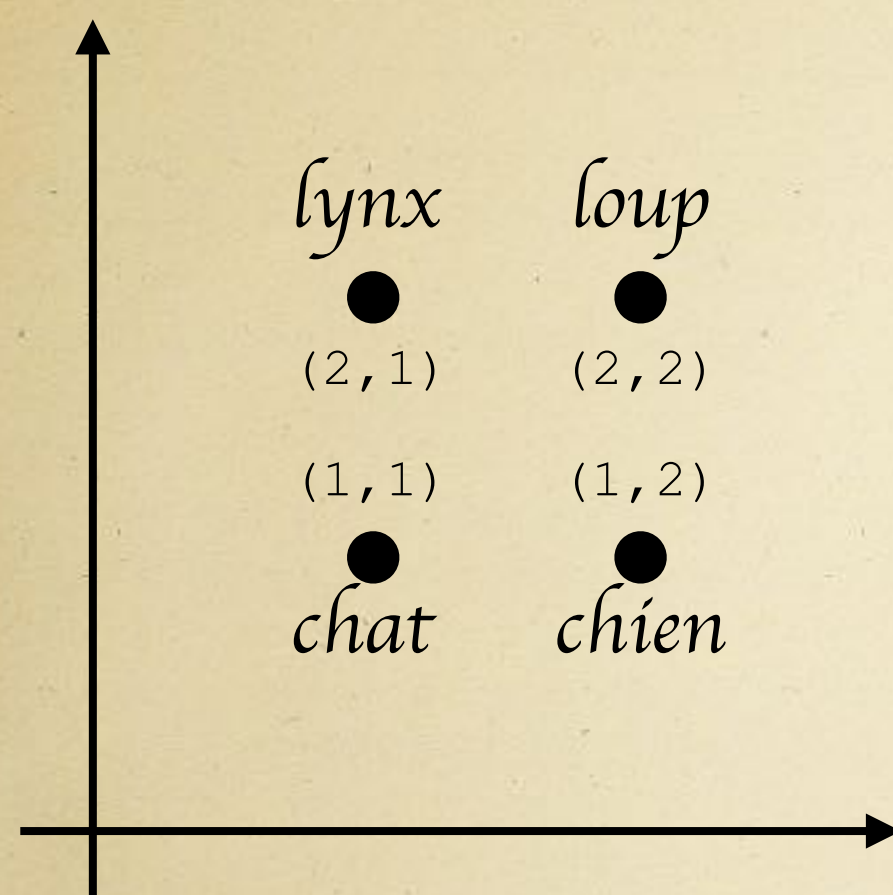
# Traduction d'un langage naturel



*Besoin des nombres représentant le sens relatif des mots.*

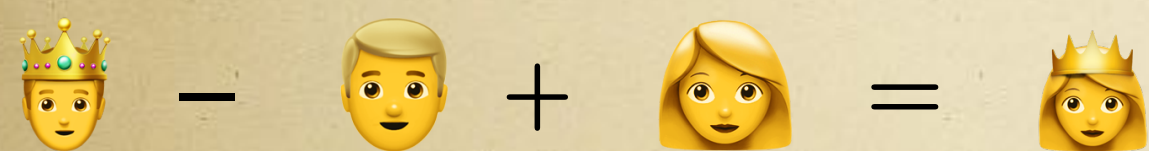
①

# Traduction d'un langage naturel



Procédure automatisable grâce à un réseau de neurones

- lecture d'énormément de phrases
- calcul des relations de proximité
- *Word2Vec* par Google ou *GloVe* par l'Université de Stanford
  - vecteur de dim  $\sim 300$  pour chaque token



Mot  $\rightarrow$  token permet d'enlever la barrière de la langue

Vecteur représentant la position dans l'espace des significations avec une notion de proximité lorsque deux tokens se ressemblent.

①

## Traduction d'un langage naturel

GloVe par l'Université de Stanford - 2024 Dolma, 300D vecteurs

chat	(-0.215, -0.253, 0.142, ... , -0.182, 0.126)
chien	(0.336, 0.170, 0.515, ... , -0.003, -0.227)
lynx	(0.022, -0.260, -0.399, ... , 0.025, -0.252)
loup	(-0.268, 0.472, 0.244, ... , -0.566, -0.404)
citrouille	(-1.470, 1.269, 0.114, ... , -0.835, -0.030)

$$\delta(\text{chat}, \text{chien}) = 8.7479$$

$$\delta(\text{chat}, \text{lynx}) = 8.2495$$

$$\delta(\text{chien}, \text{loup}) = 6.7051$$

$$\delta(\text{lynx}, \text{loup}) = 8.0046$$

$$\delta(\text{chat}, \text{loup}) = 9.1002$$

$$\delta(\text{chat}, \text{citrouille}) = 13.5457$$

②

## Modèle de fondation et spécialisation vers un chatbot

Cœur des LLM : *prédire le prochain token*

Exemple :

- GPT (Generative Pre-training Transformer)

Modèle d'intelligence artificielle

Issu d'apprentissage auto-supervisé.

- ChatGPT

Produit commercial d'OpenAI

Ne peut pas se contenter de chercher que les tokens suivants.

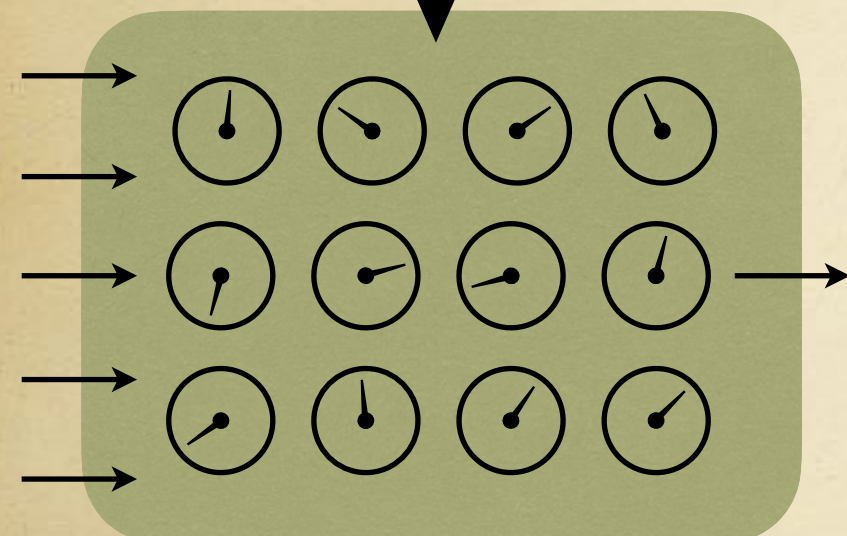
Notion de continuité dans la « conversation ».



②

## Étape 1 : pre-training

textes sur Internet



modèle de fondation

Modèle de fondation : Un modèle d'IA qui est entraîné sur une certaine tâche générique mais dans le but d'être ensuite adapté à d'autres tâches plus spécifiques.

Tâche : *prédire le prochain token*

Complétion automatique : une suite possible, de façon crédible, plausible, sans notion de vérité.

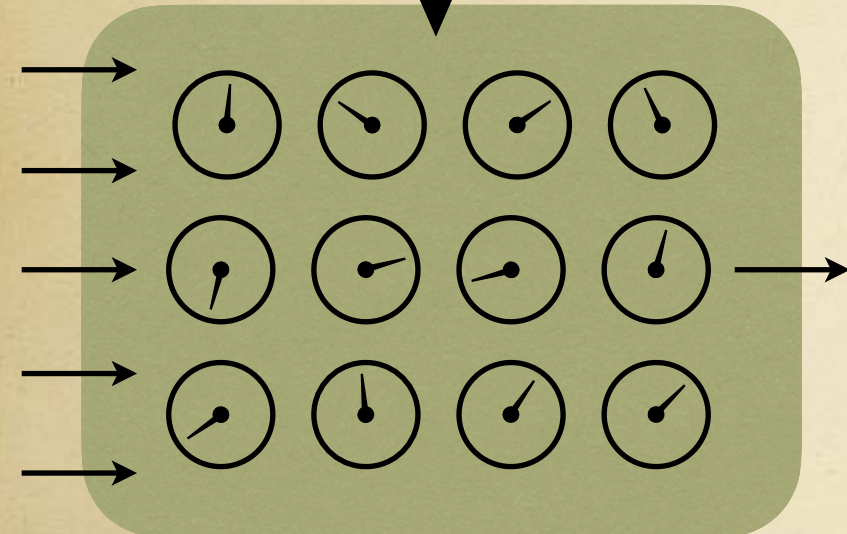
NextTokenPrediction

twinkle twinkle, little star how I wonder what you are up above the world so high like a diamond in the sky

②

# Étape 1 : pre-training Apprentissage auto-supervisé

textes sur Internet



modèle de fondation

Grandes quantités de données étiquetées par l'humain -> irréaliste

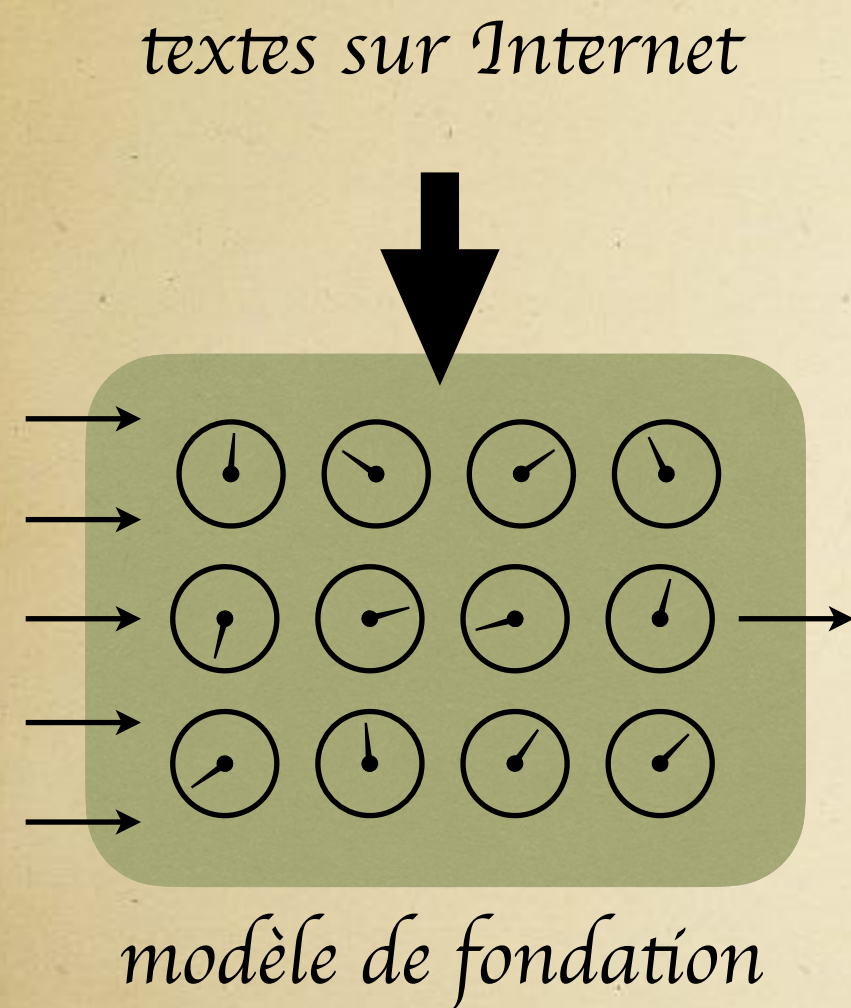
Le modèle apprend à partir des données brutes.

Le	modèle
Le modèle	apprend
Le modèle apprend	à
Le modèle apprend à	partir
Le modèle apprend à partir	des
Le modèle apprend à partir des	données
Le modèle apprend à partir des données	brutes.

②

## Étape 1 : pre-training

Un transformeur lit des dizaines de milliards de phrases et permet de prédire des tokens.



Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

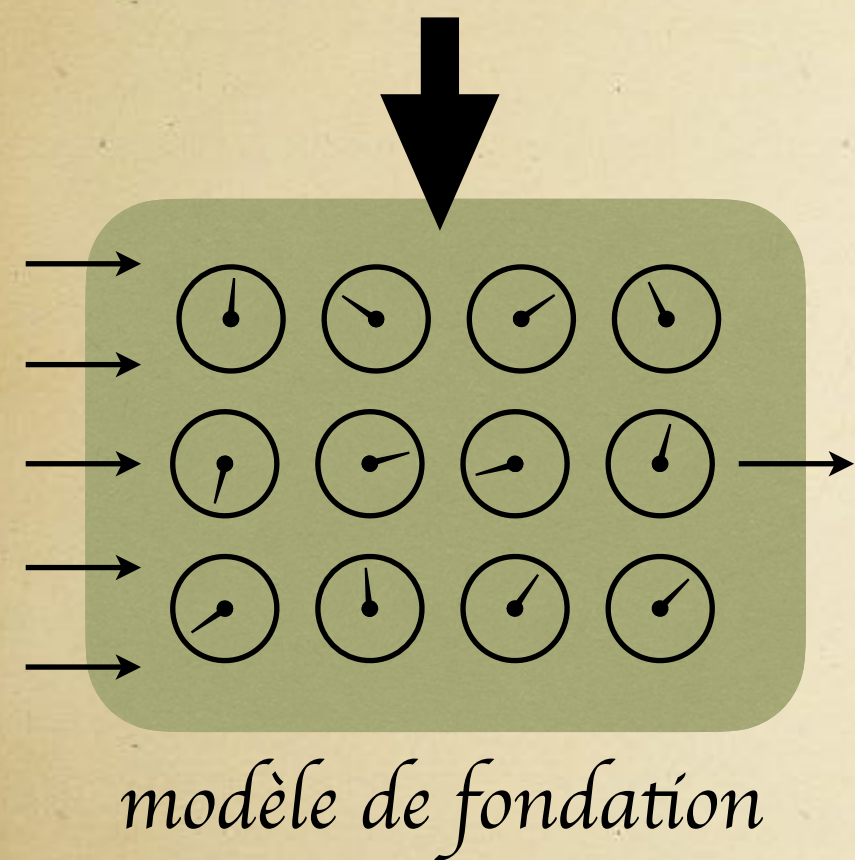
**Table 2.2: Datasets used to train GPT-3.** “Weight in training mix” refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets are seen less than once.

*super auto-complète, grammaire, syntaxe, règles, concepts*

②

## Étape 1 : pre-training

textes sur Internet



### *Reproduction des biais*

- *discriminatoires (raciste, sexiste, homophobe, religieux, etc.)*
  - *cognitifs (confirmation, autorité, etc.)*
  - *médiatiques (sélection, langage, etc.)*
- qui ont servi à leur construction.*

②

## Étape 1 : pre-training - utilisation en pratique

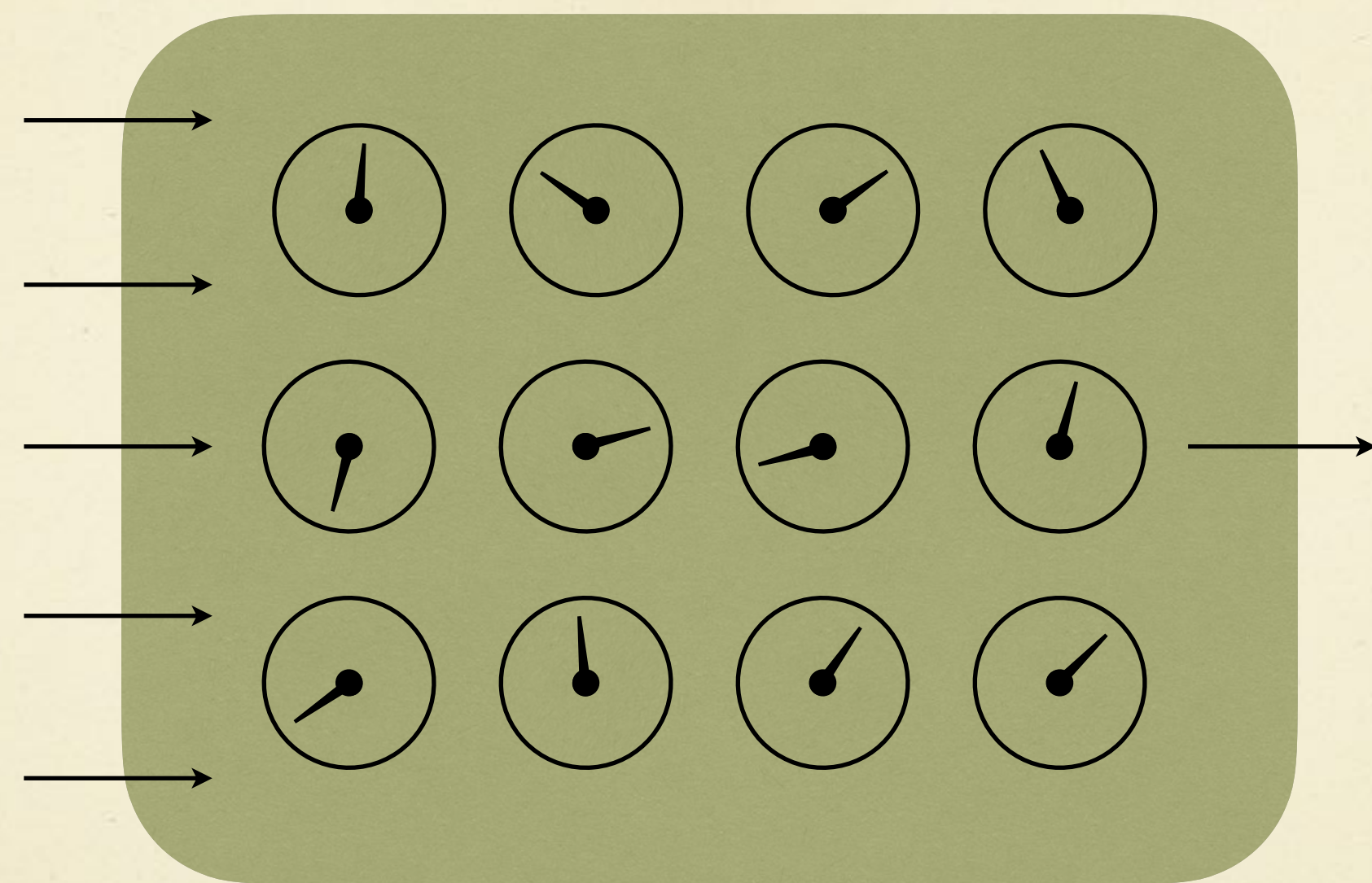
fenêtre de contexte

GPT4

~32000 tokens

~25000 mots

~100 pages



vecteur de probabilité  
de tokens

- sortie est redonné en entrée pour compléter un texte
- GPT ne répond pas à une question, il complète le texte
- modification de la sortie en choisissant bien le prompt
- notion de préprompt (position géographique, date, heure, etc.)
- préprompt non connu pour ChatGPT (secret, évolution)

②

## Étape 1 : pre-training - utilisation en pratique

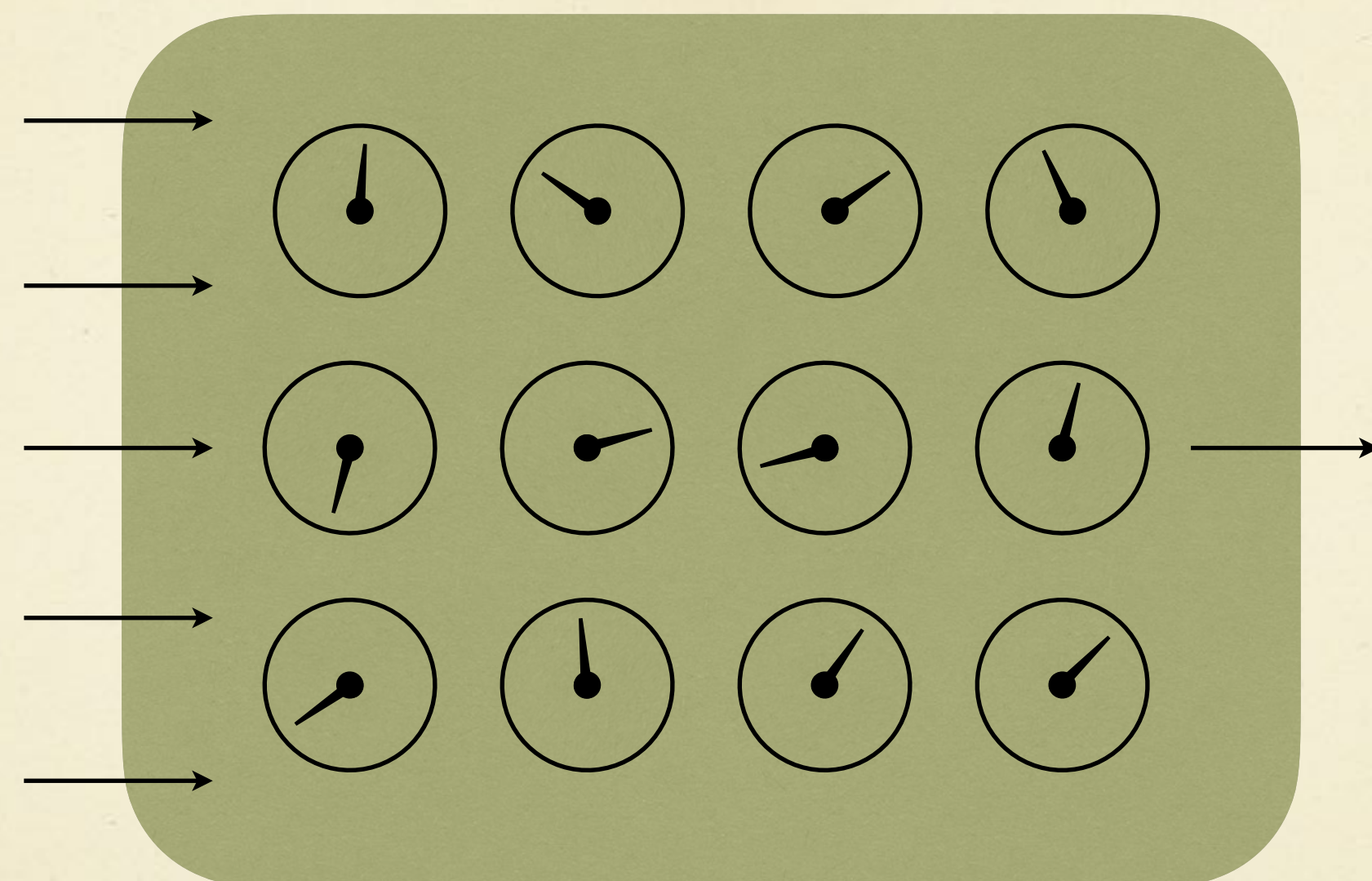
fenêtre de contexte

GPT4

~32000 tokens

~25000 mots

~100 pages



vecteur de probabilité  
de tokens

- instructions invisibles pour l'utilisateur avec un biais de l'IA
- ex : Grok, l'IA d'Elon Musk - 14.05.2025 **Le Monde**  
thèse d'un « génocide blanc » en Afrique du Sud
- ex : Qwen - les dates clés qui ont conduit  
aux manifestations de la place Maïdan / Tian'anmen

②

## Étape 2 : fine-tuning supervisé

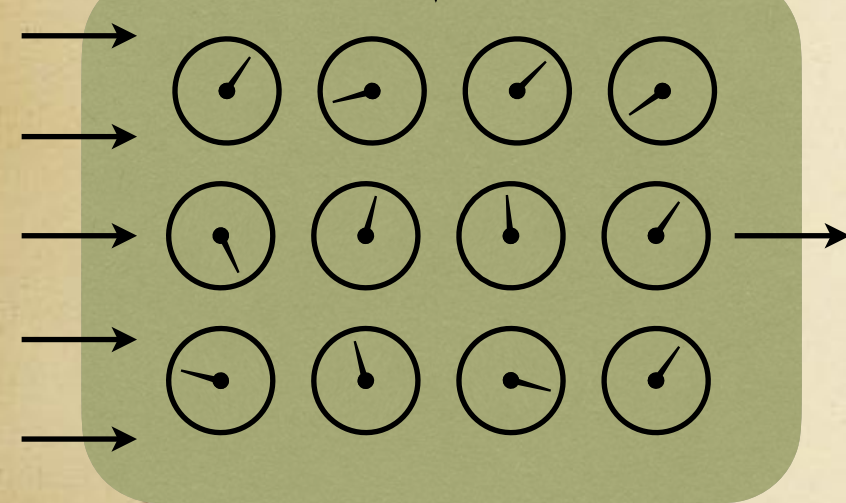
Modèle instruct : Modèle entraîné en apprentissage supervisé avec des exemples écrits style « conversation entre un utilisateur humain et un chatbot aimable ».

questions / réponses objectifs



-> Datasets

données annotées  
questions/réponses

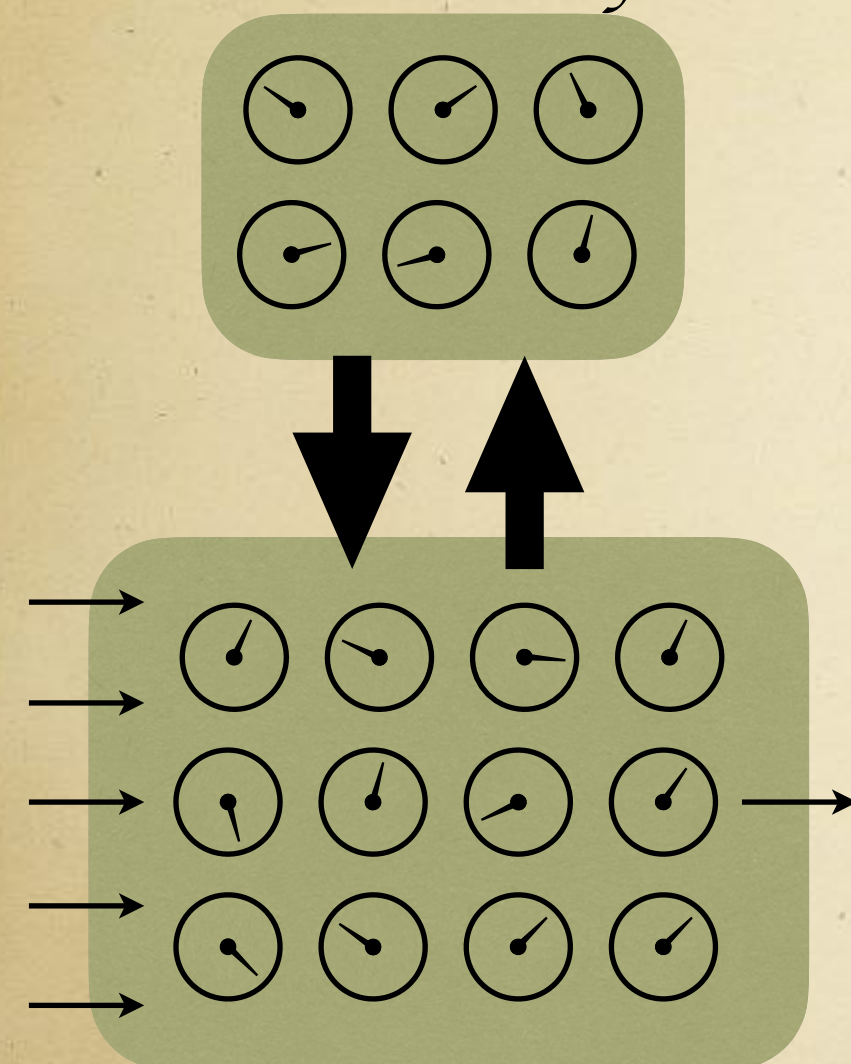


modèle instruct

②

## Étape 3 : fine-tuning par les préférences

modèle de récompense



DPO (Direct Preference Optimisation) : retour des utilisateurs

Préférences intellectuelles



Contenus non désirables (dangereux, incitations haineuses, idées suicidaires, manquement d'éthiques, etc.)

**TIME**

Données insuffisantes -> création d'un modèle de récompense (humain virtuel)

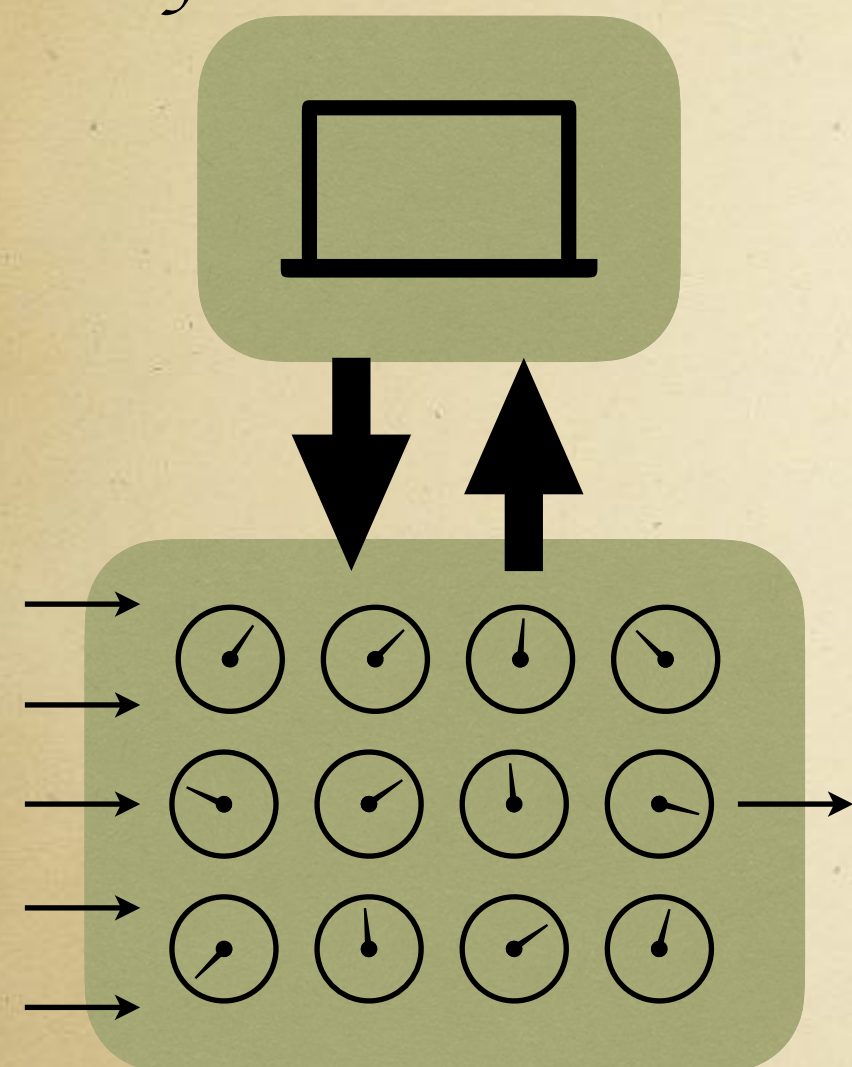
Interpolation réussie mais reste des failles (articles inventés, instabilité, etc.)

②

## Étape 4 : fine-tuning par le raisonnement

Chain of thoughts :

vérification automatisée



- raisonnement sur un brouillon (réflexion)
- réponses extraites
- apprentissage par renforcement automatisé en utilisant ses propres réponses sur des problèmes vérifiables (sources, codes, etc.)

RLVR (Reinforcement Learning With Vérifiable Rewards)

Limite des données de l'humanité (énergie fossile)

Données synthétiques -> pas réaliste

Apprentissage par renforcement (énergie renouvelable)



Ilya Sutskever

<i>Magie ?</i>	<i>GPT3</i>	<i>GPT3.5</i>	<i>GPT4</i>	<i>GPT5</i>	<i>Mistral Large 2</i>	<i>DeepSeekV3</i>
<i>Paramètres</i>	<i>175 milliards</i>	<i>-</i>	<i>1,8 trillion</i>	<i>-</i>	<i>123 milliards</i>	<i>671 milliards</i>
<i>Nombre de tokens disponibles</i>	<i>50K</i>	<i>100K</i>	<i>100K</i>	<i>100K</i>	<i>128K</i>	<i>128K</i>
<i>Fenêtre de contexte</i>	<i>2048</i>	<i>4096 / 16K</i>	<i>8K / 32K</i>	<i>-</i>	<i>128K</i>	<i>128K</i>
<i>Nombre de tokens d'entraînement</i>	<i>300 milliards</i>	<i>-</i>	<i>13 trillions</i>	<i>-</i>	<i>1 trillion</i>	<i>14,8 trillions</i>

②

## Étape 5 : et les autres améliorations

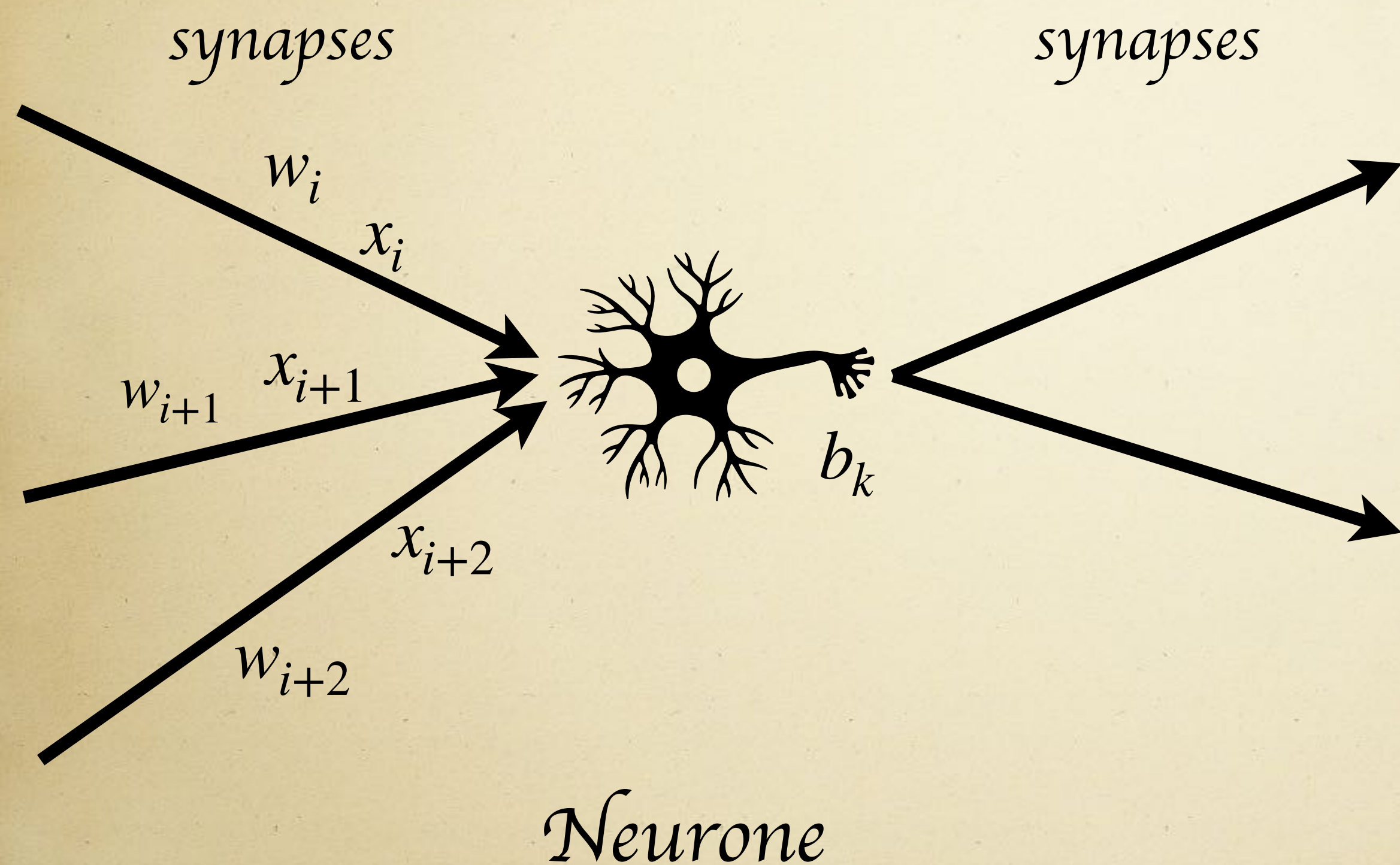
Problème de la date limite d'entraînement :

- recherche sur Internet avec synthèse
- exécution du code Python
- accord avec certaines presses
- ...

③

# Réseaux de neurones artificiels profonds

Ensemble de neurones virtuels disposés en réseau virtuel



- intensité des signaux  $x_i$
- poids des synapses  $w_i$ 
  - $w_i > 0$  synapse activatrice
  - $w_i < 0$  synapse inhibitrice
  - $w_i = 0$  synapse au repos
- biais du neurone  $b_k$
- fonction d'activation  $A$

$$X = \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

$$W = \begin{pmatrix} w_0 \\ \vdots \\ w_{n-1} \end{pmatrix}$$

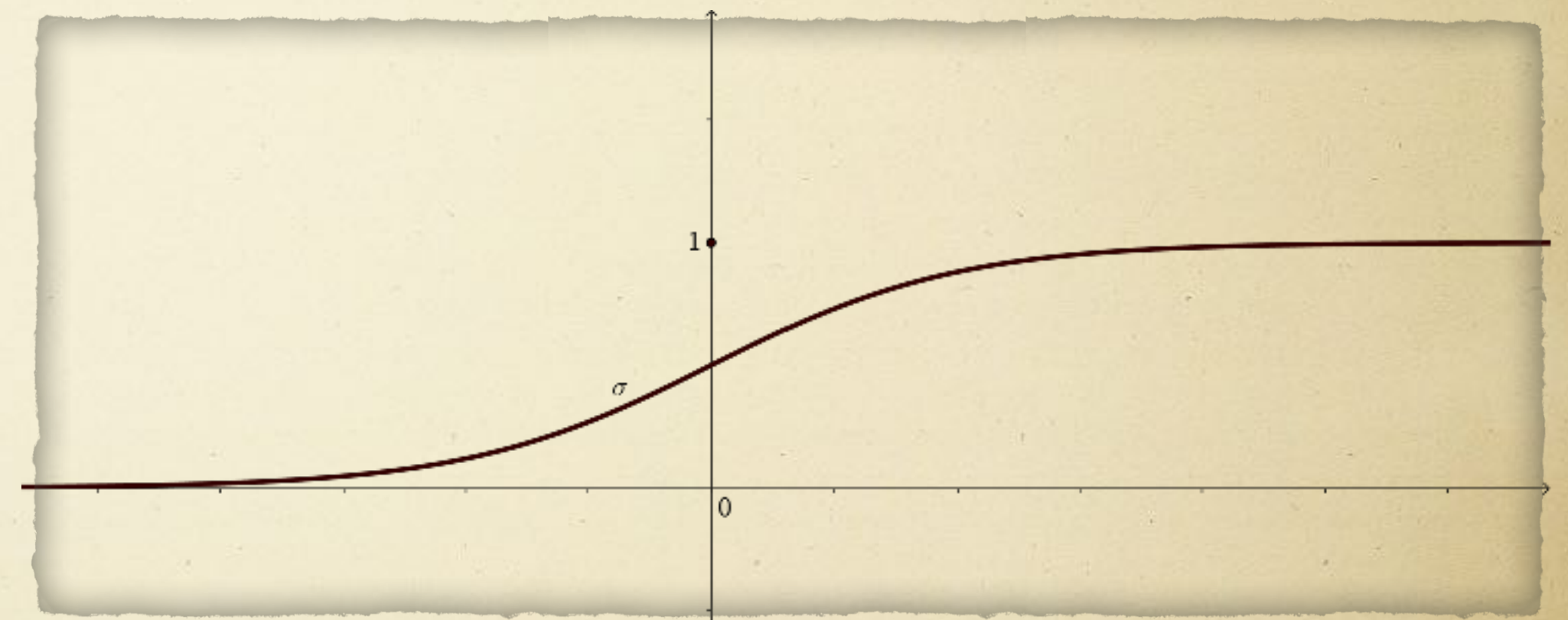
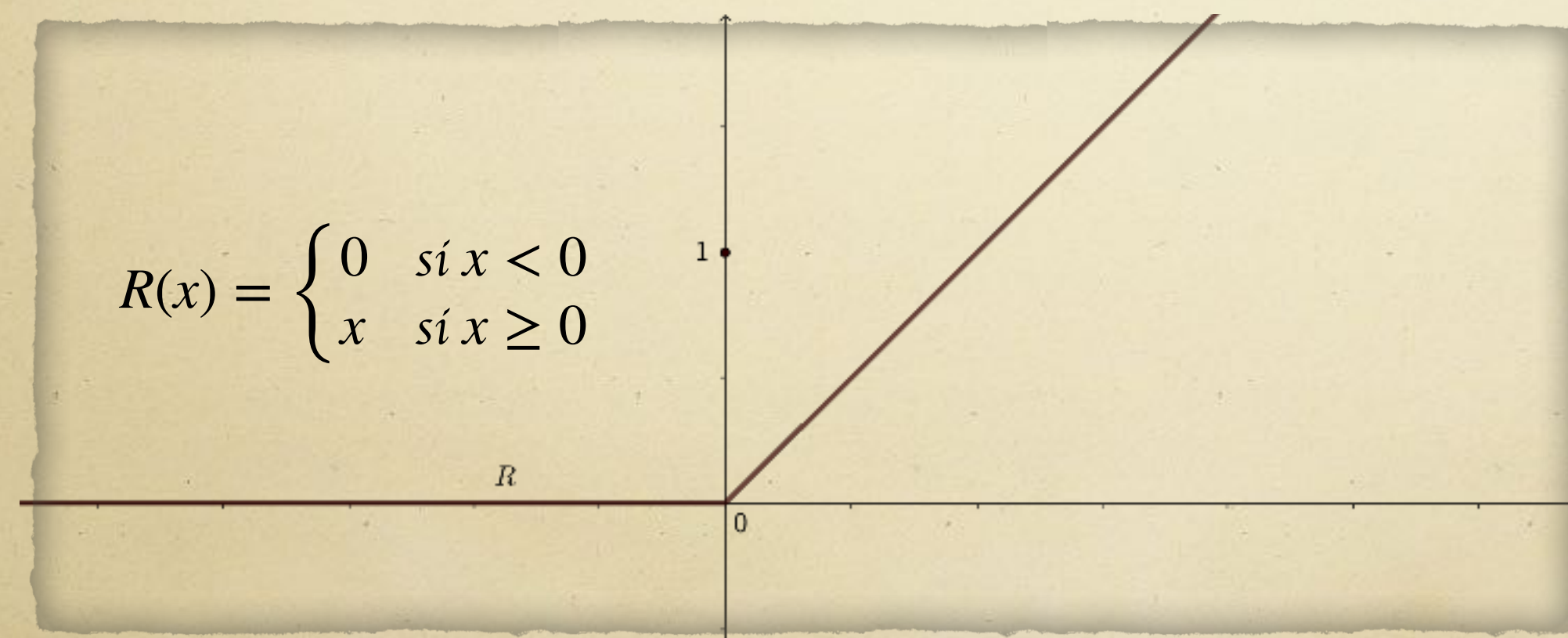
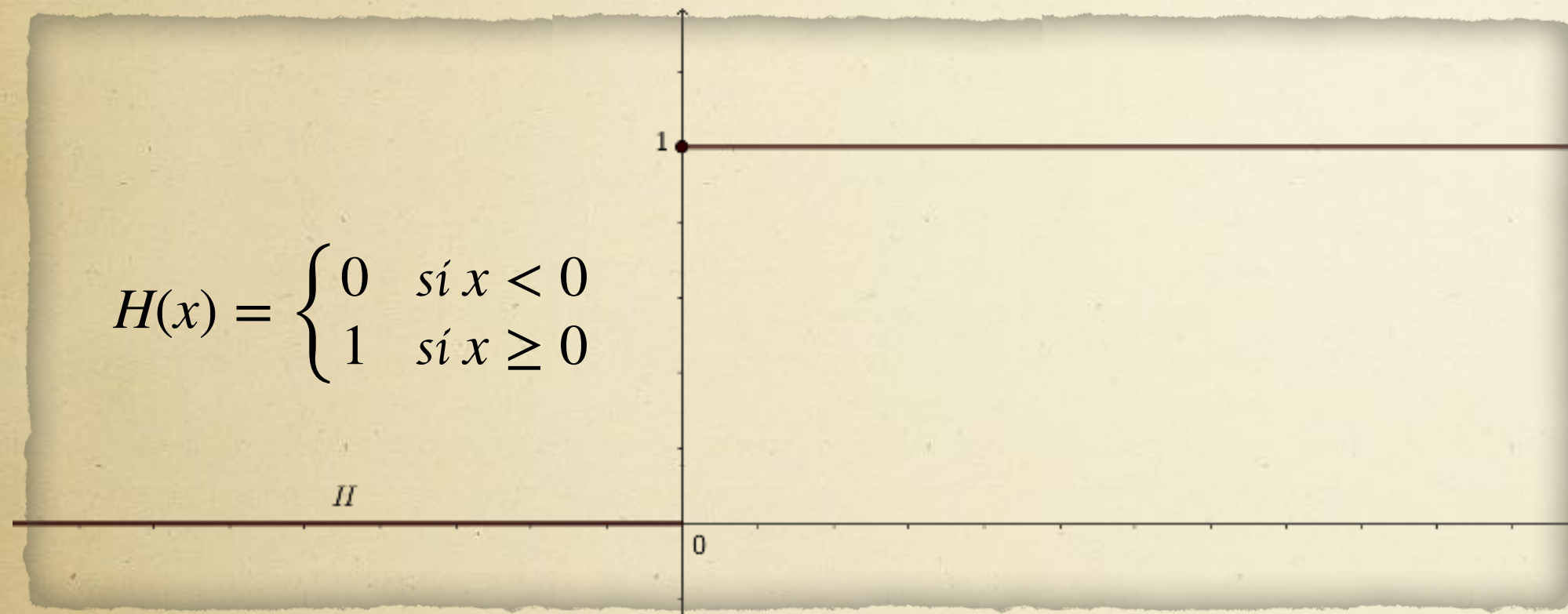
Un neurone s'active ou pas en fonction des informations reçues :  $A({}^tWX + b_k)$

③

## Réseaux de neurones - fonction d'activation

Un neurone s'active ou pas en fonction des informations reçues :  $A({}^tWX + b_k)$

Heaviside, ReLU, sigmoïde, tangente, hyperbolique, max-pool, soft-max, etc.



$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad \text{et} \quad \sigma'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = \sigma(x)(1 - \sigma(x))$$

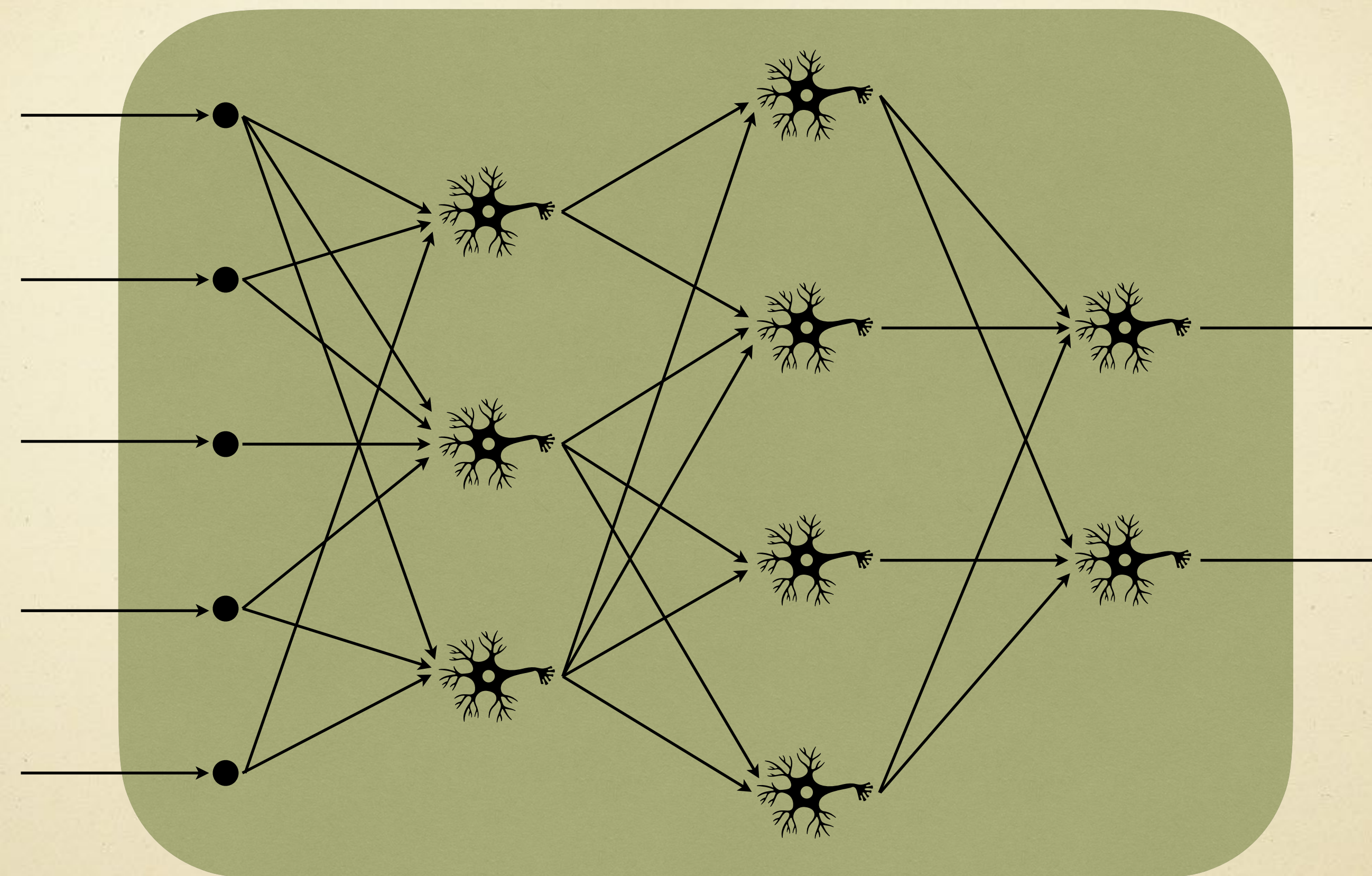
Un neurone sigmoïde s'active entre 0% et 100%, adapté aux probabilités attendues.

③

# Réseaux de neurones

Données brutes :

- amplitude d'un son
- couleurs RGB des pixels d'une image
- vecteurs de proximité des tokens dans un LLM



données brutes

1<sup>ère</sup> couche

2<sup>nd</sup> couche

couche de sortie

Niveau  
d'activation des  
neurones finaux  
donne le vecteur de  
probabilité

✓ réseaux feedforward

✗ réseaux récurrents

③

## Théorème d'approximation universelle

Soit  $f : [0,1]^n \rightarrow [0,1]^m$ .  $\forall \varepsilon > 0, \exists$  un réseau de neurone  $RN$  tel que  $\|f - RN\| < \varepsilon$ .

*Les réseaux de neurones peuvent approximer toute fonction continue sur un intervalle compact.*

*Turing 1950 - Computing Machinery and intelligence.*

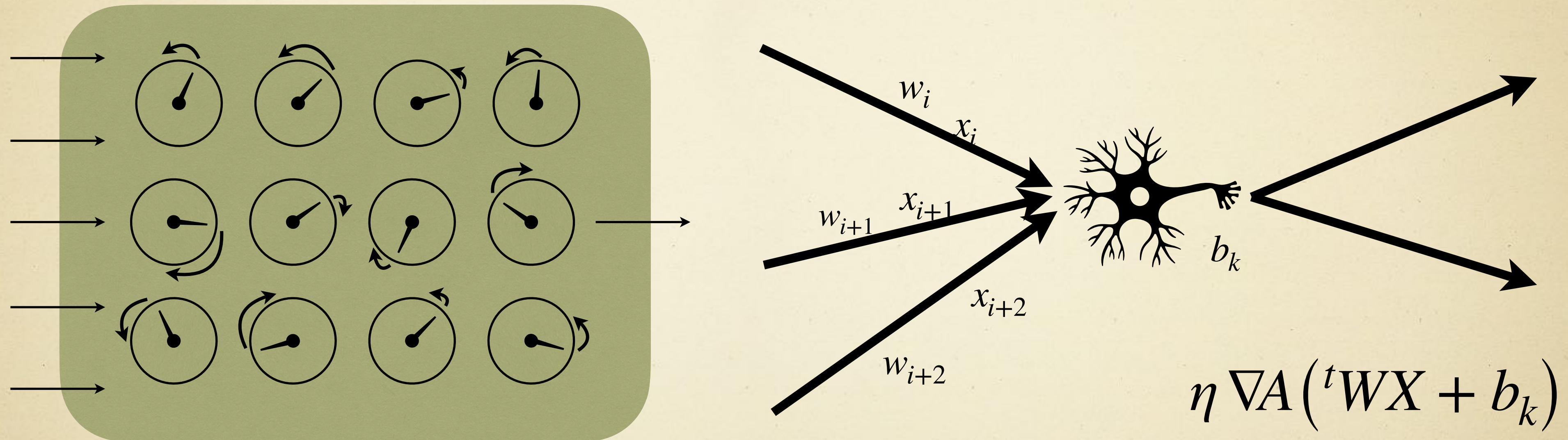
« It is possible to have a machine which can be made to alter its own instructions. Suppose that a machine is made in this way and that its initial state is such that its behaviour is predictable. It will then be possible to find some way of feeding it with an input which will alter its instructions in such a way that they will no longer be predictable by us in advance. Such a machine will be said to be 'supercritical'. »

« Estimates of the storage capacity of the brain vary from  $10^{10}$  to  $10^{15}$  binary digits. »

③

# Réseaux de neurones - descente de gradient stochastique

Ajuster les paramètres grâce au gradient de la fonction de perte.

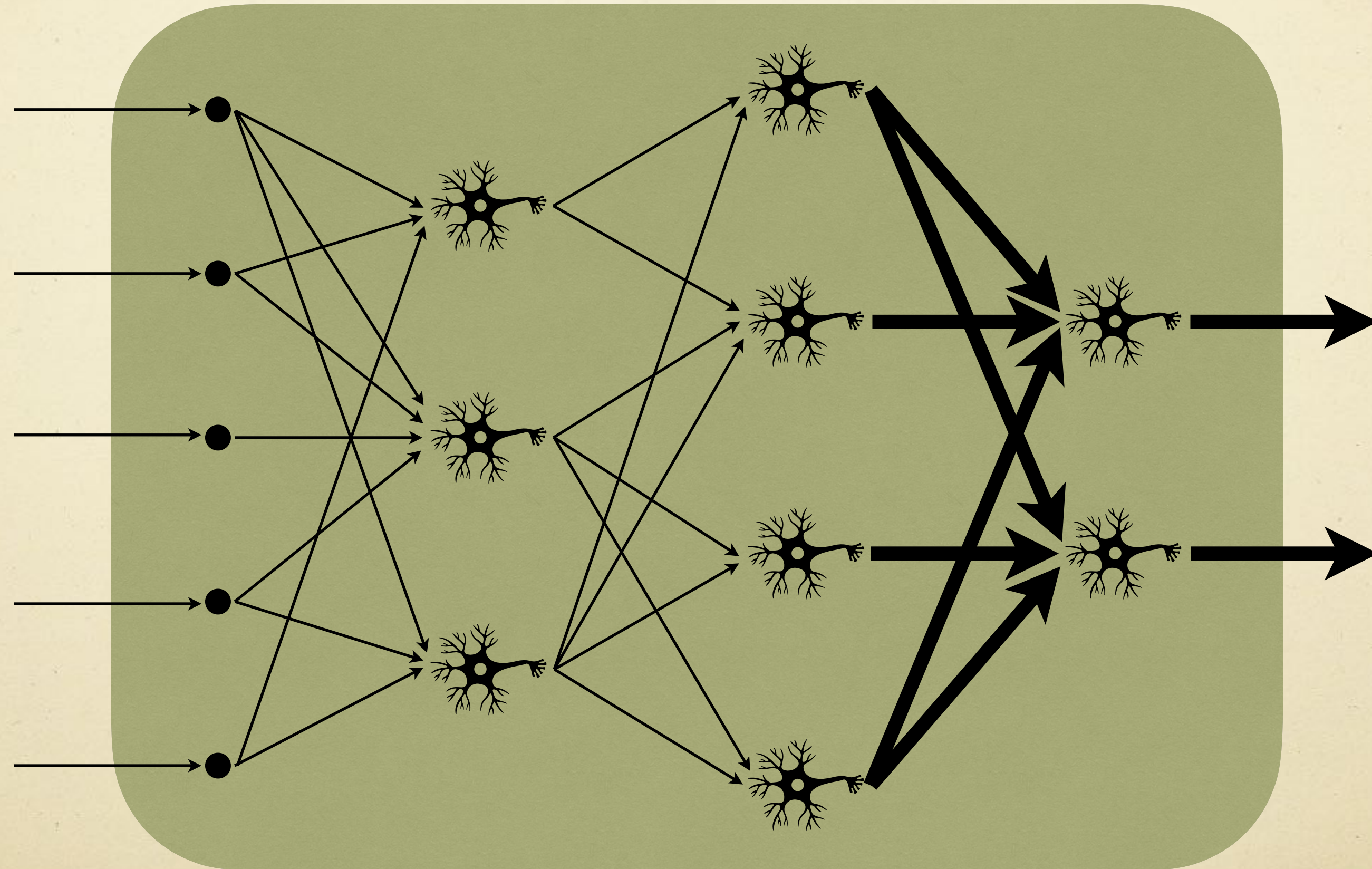


- choix de quelques données au hasard
- calcul des ajustements à faire grâce au gradient
- rétropropagation de l'information

③

# Réseaux de neurones - descente de gradient stochastique

Ajuster les paramètres grâce au gradient de la fonction de perte.

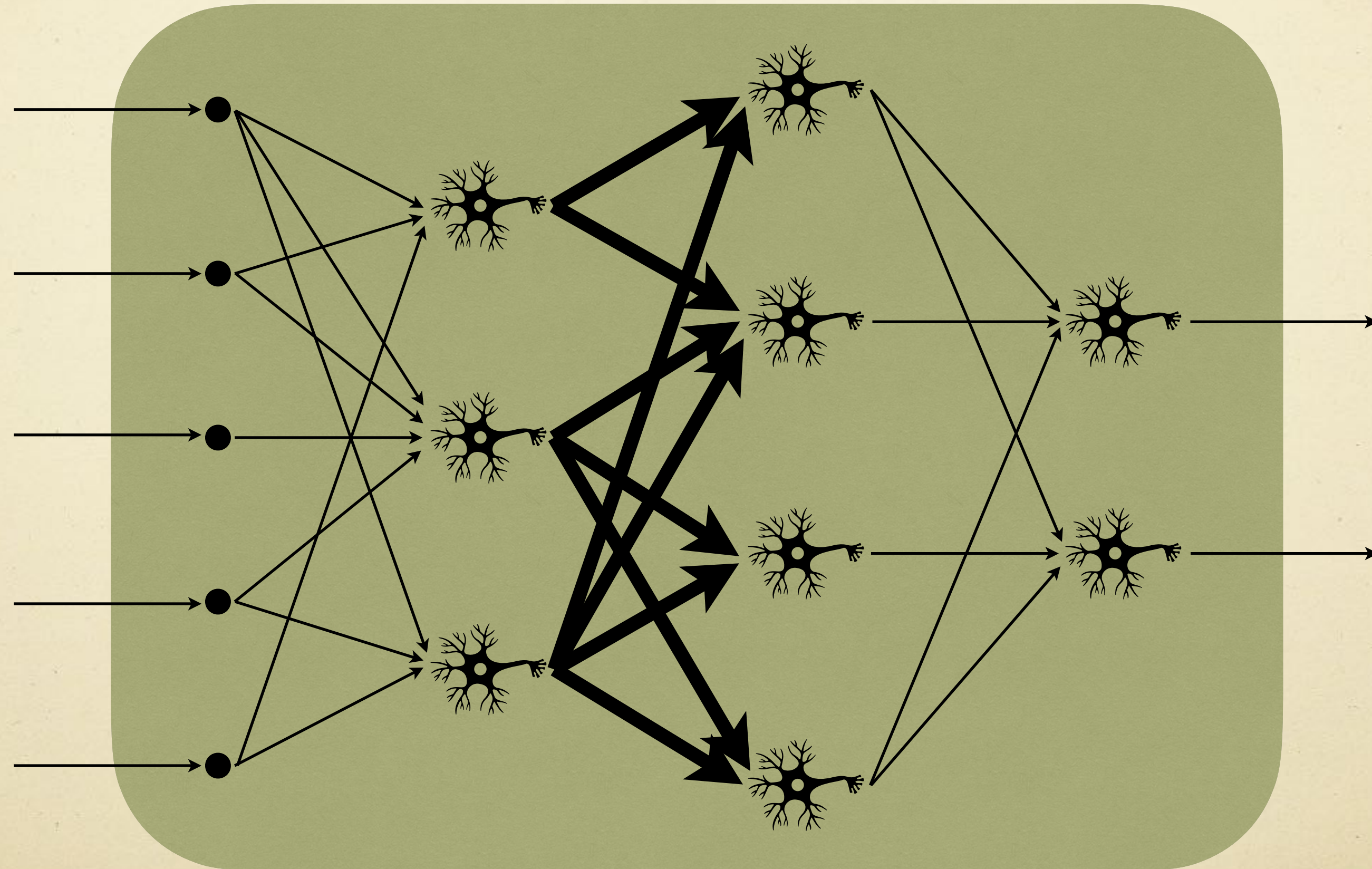


$$\eta \nabla A (^tWX + b_k)$$

③

# Réseaux de neurones - descente de gradient stochastique

Ajuster les paramètres grâce au gradient de la fonction de perte.

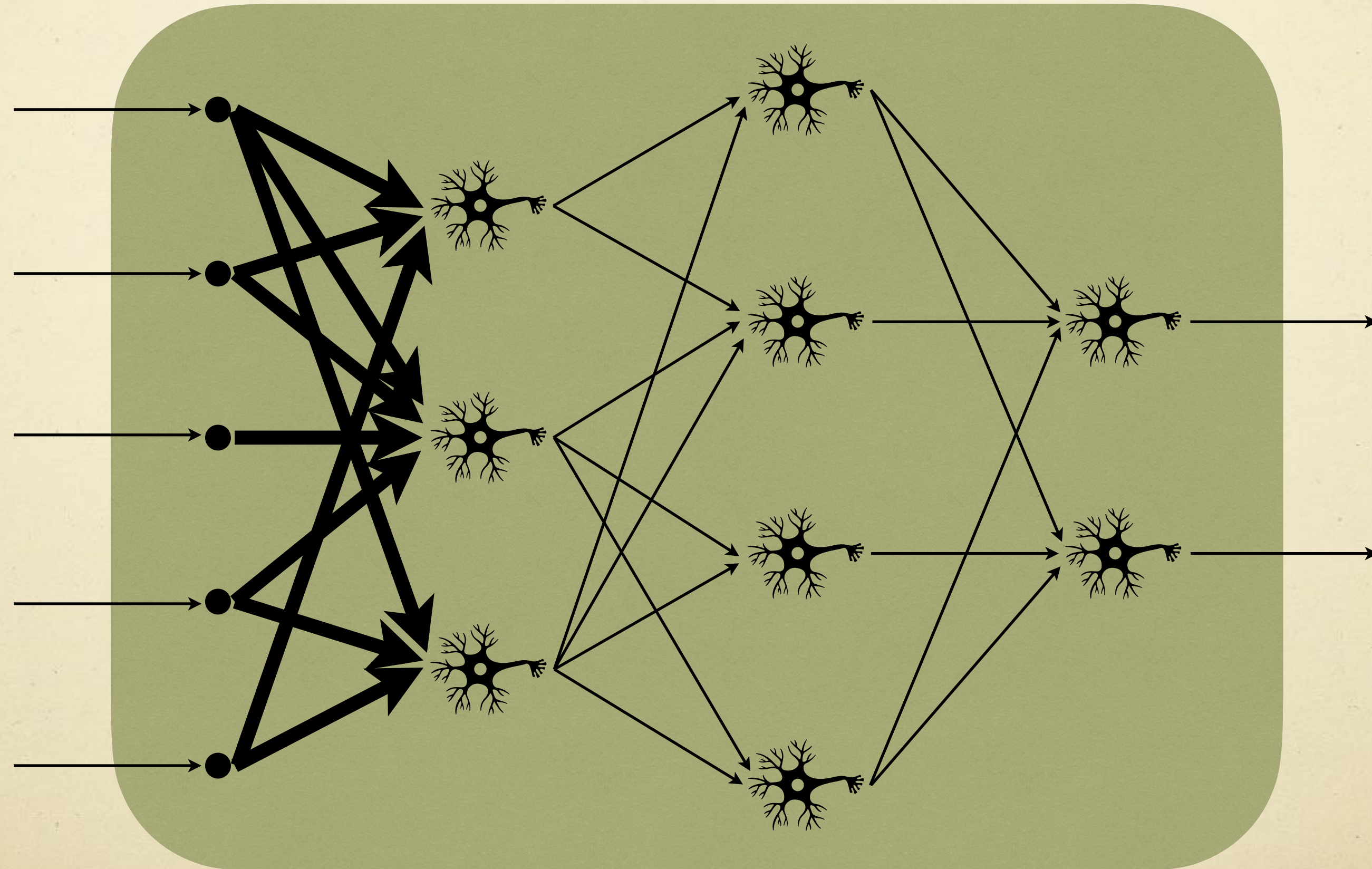


$$\eta \nabla A ({}^tWX + b_k)$$

③

# Réseaux de neurones - descente de gradient stochastique

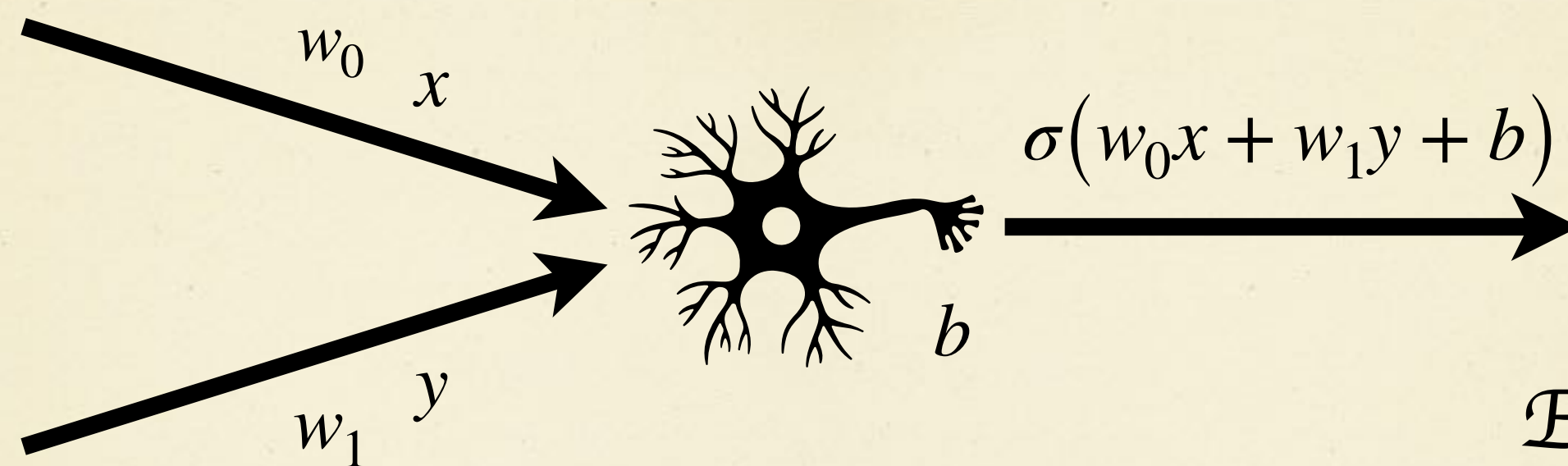
Ajuster les paramètres grâce au gradient de la fonction de perte.



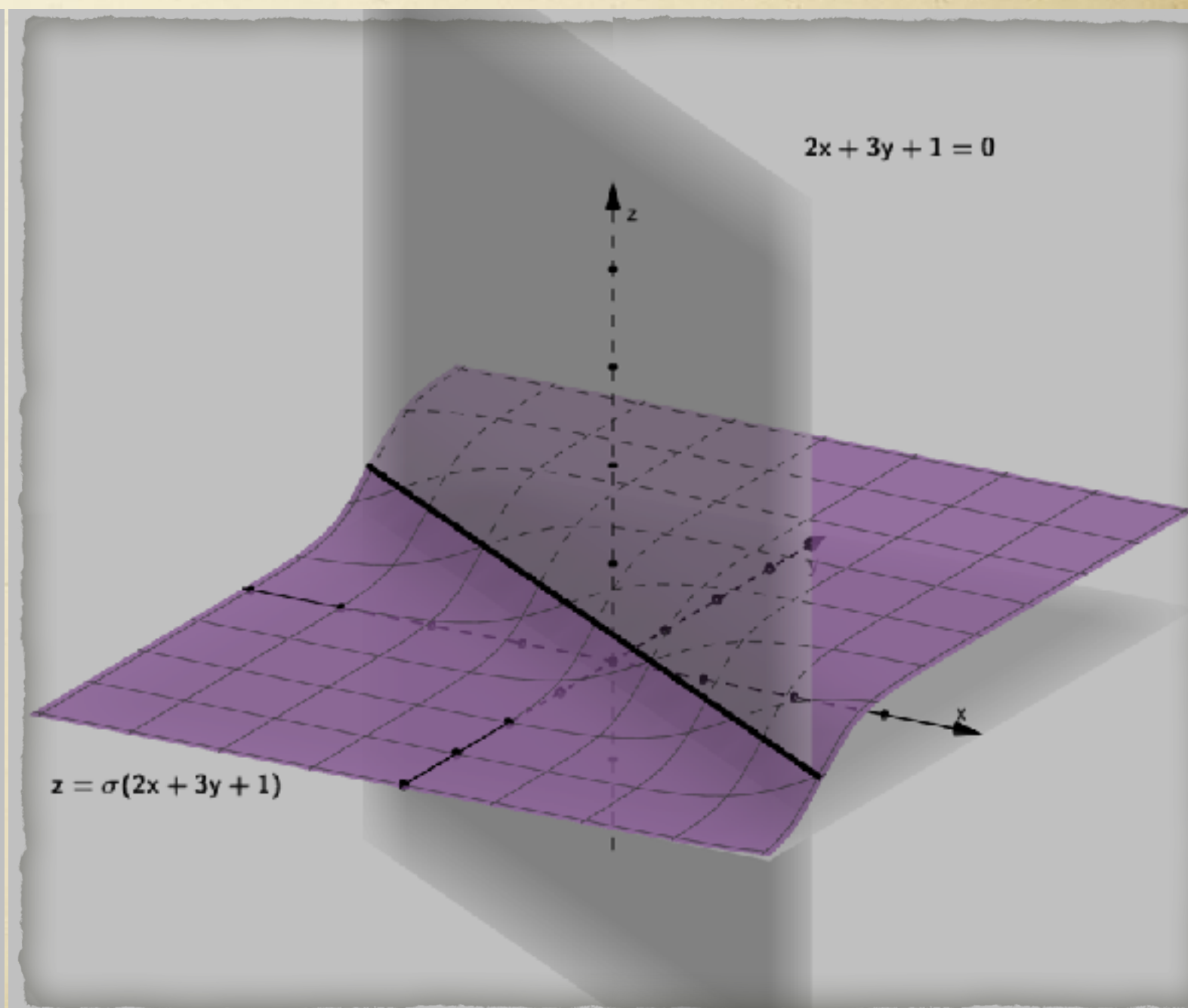
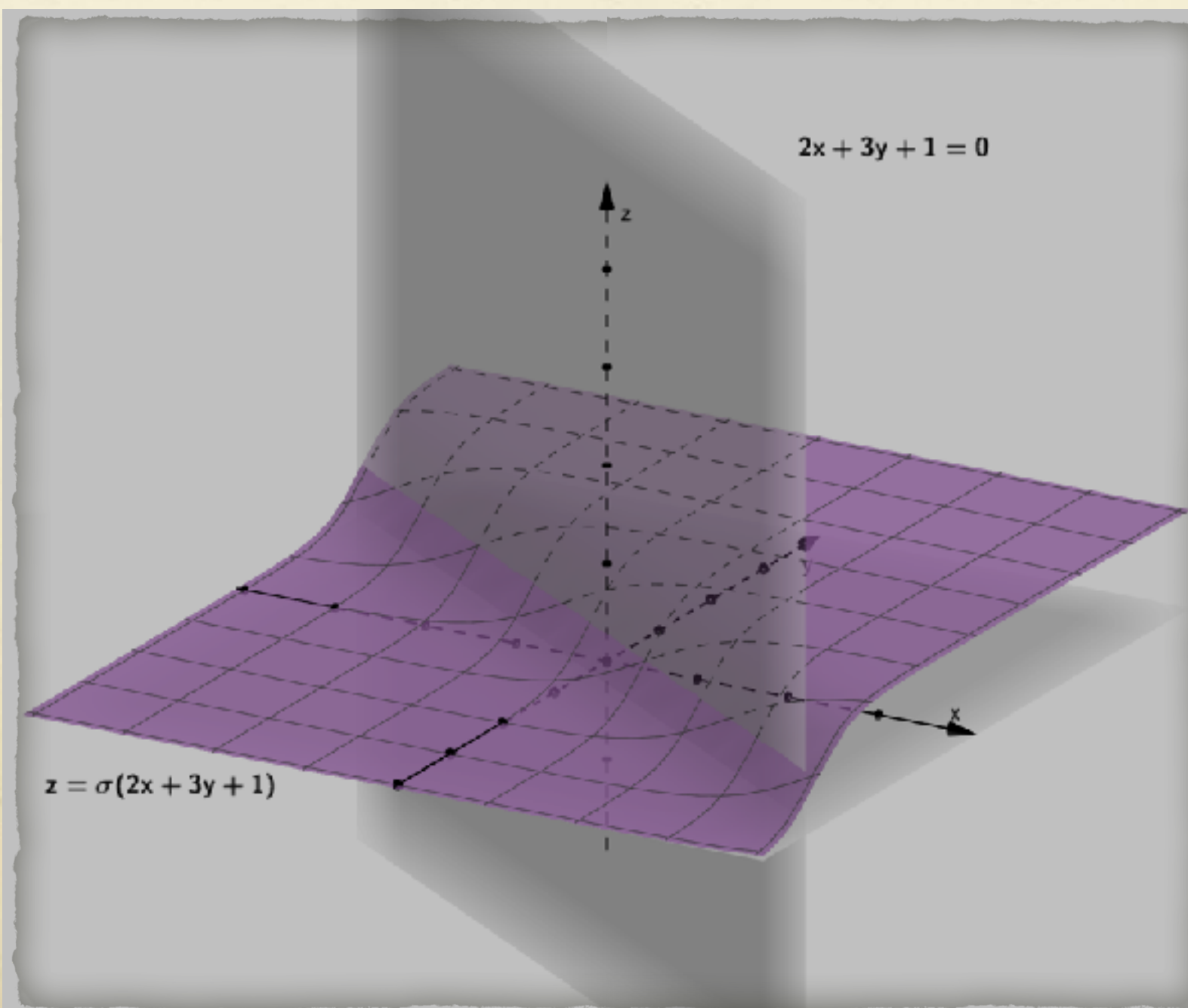
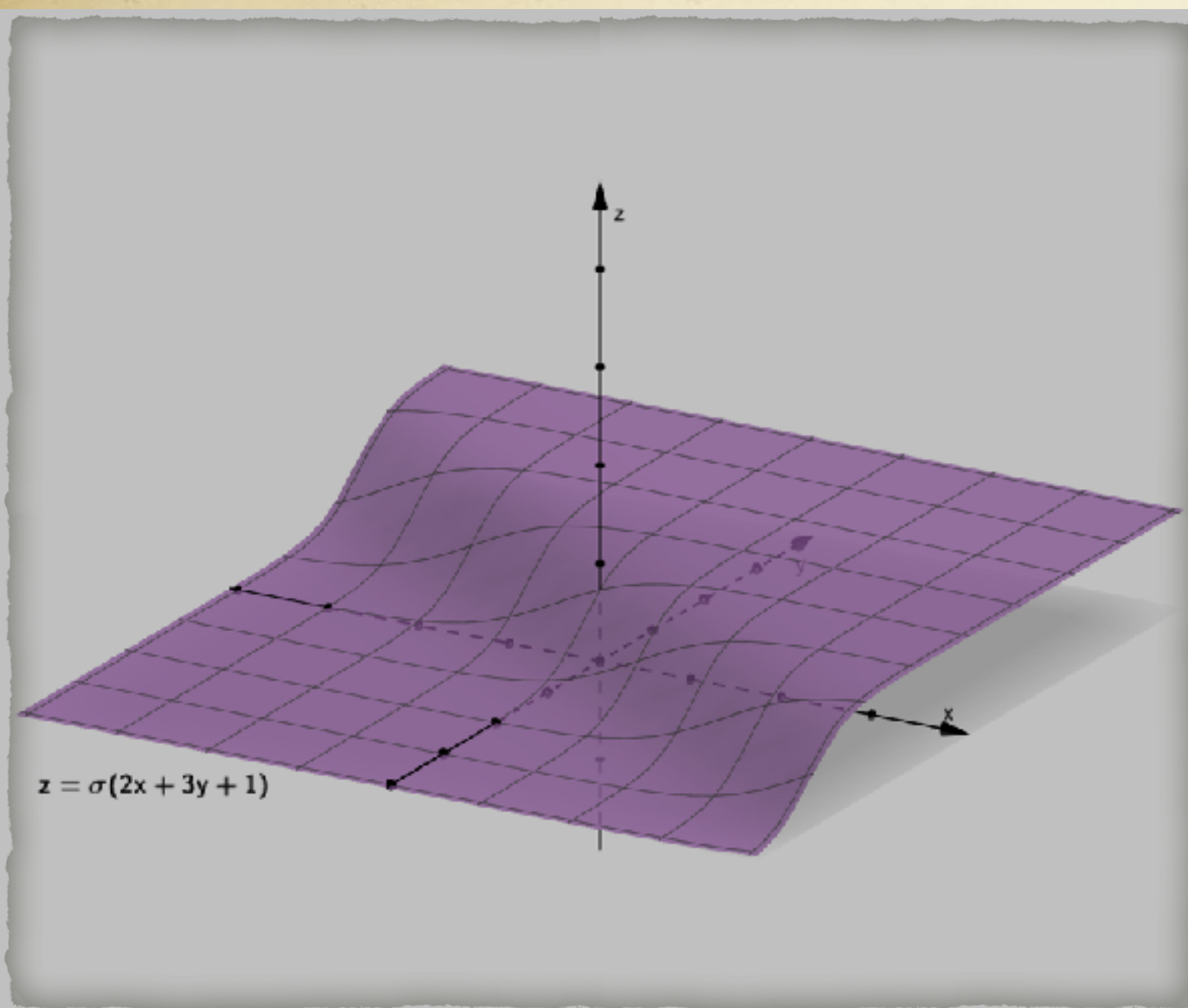
$$\eta \nabla A ({}^tWX + b_k)$$

④

# TP : Perceptron affine



Exemple :  $w_0 = 2, w_1 = 3, b = 1$

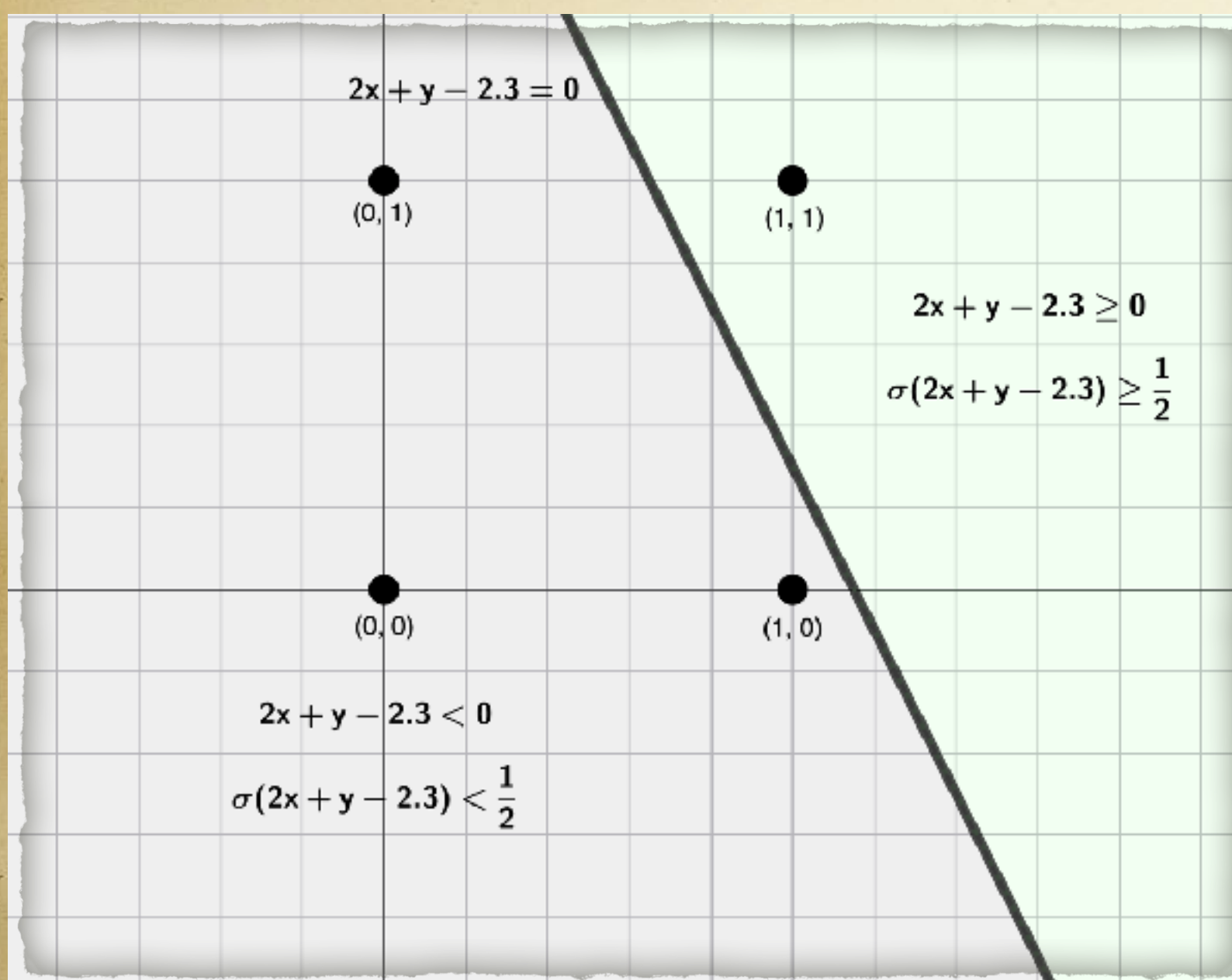


④

# TP : Perceptron affine - Fonctions logiques

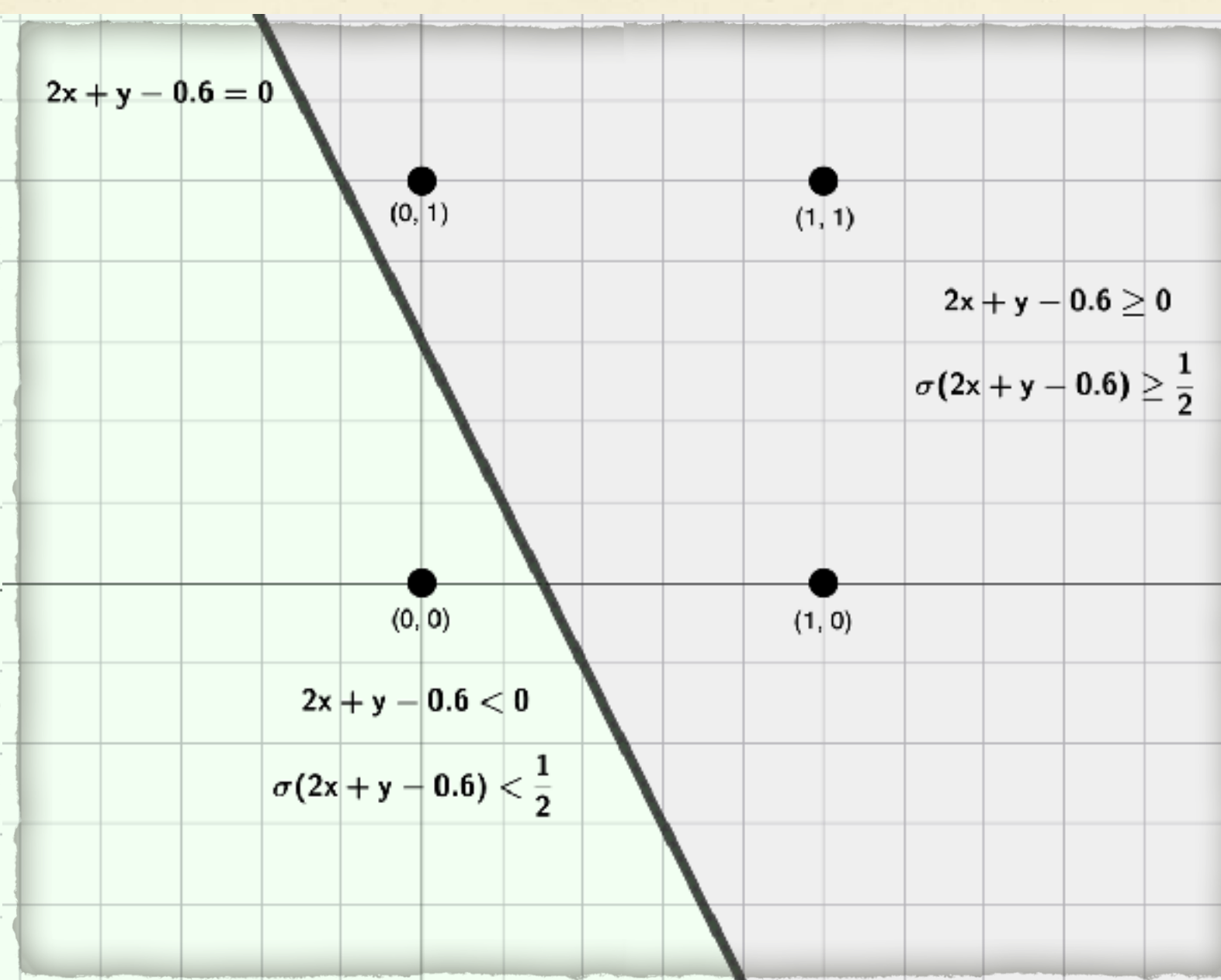
Conjonction

$$x \wedge y$$



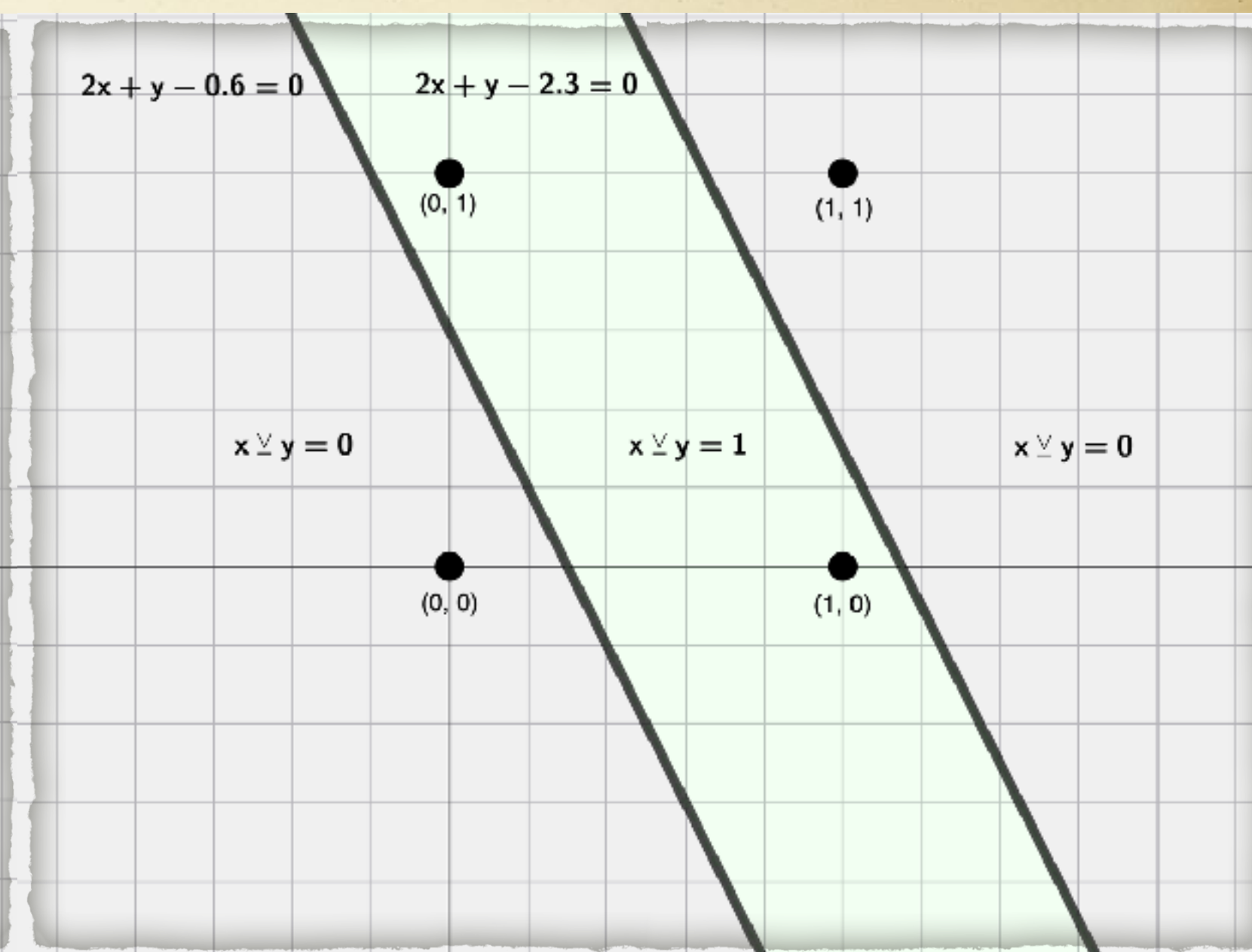
Disjonction

$$x \vee y$$



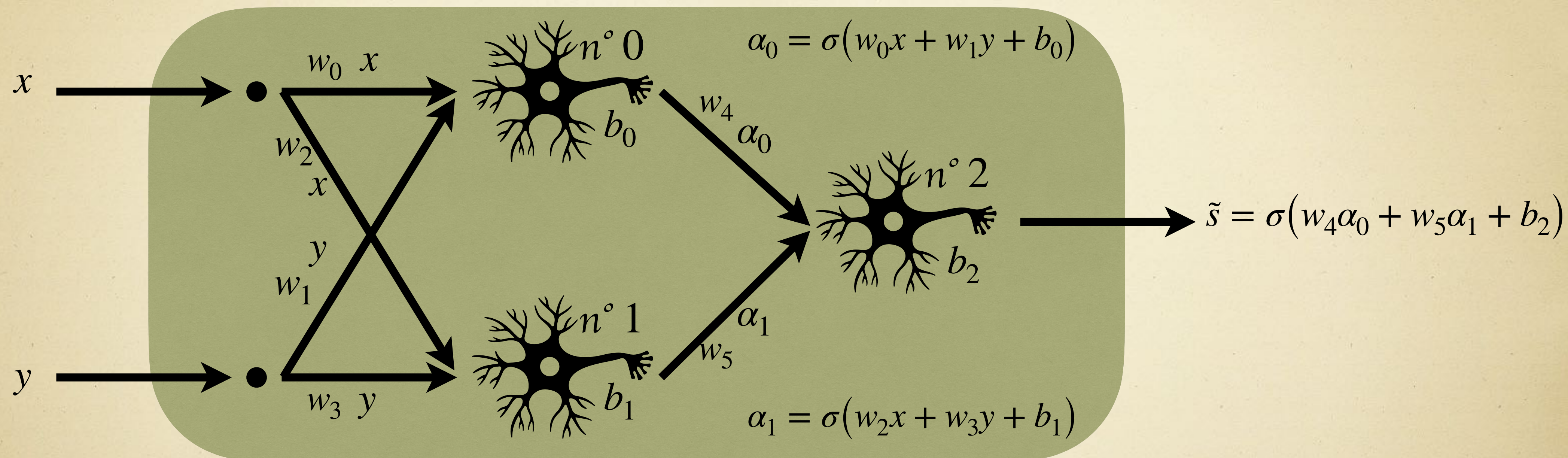
Disjonction exclusive

$$x \underline{\vee} y$$



④

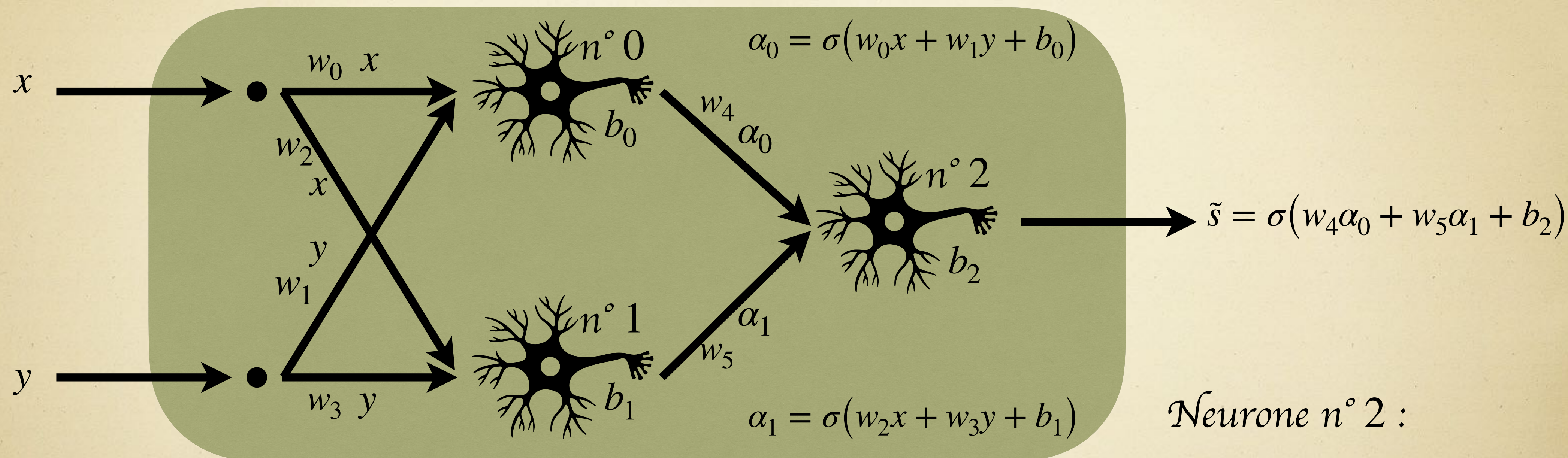
# TP : Perceptron affine - ou-exclusif



$$x \underline{\vee} y = (x \wedge \neg y) \vee (\neg x \wedge y)$$

④

# TP : Perceptron affine - rétropropagation



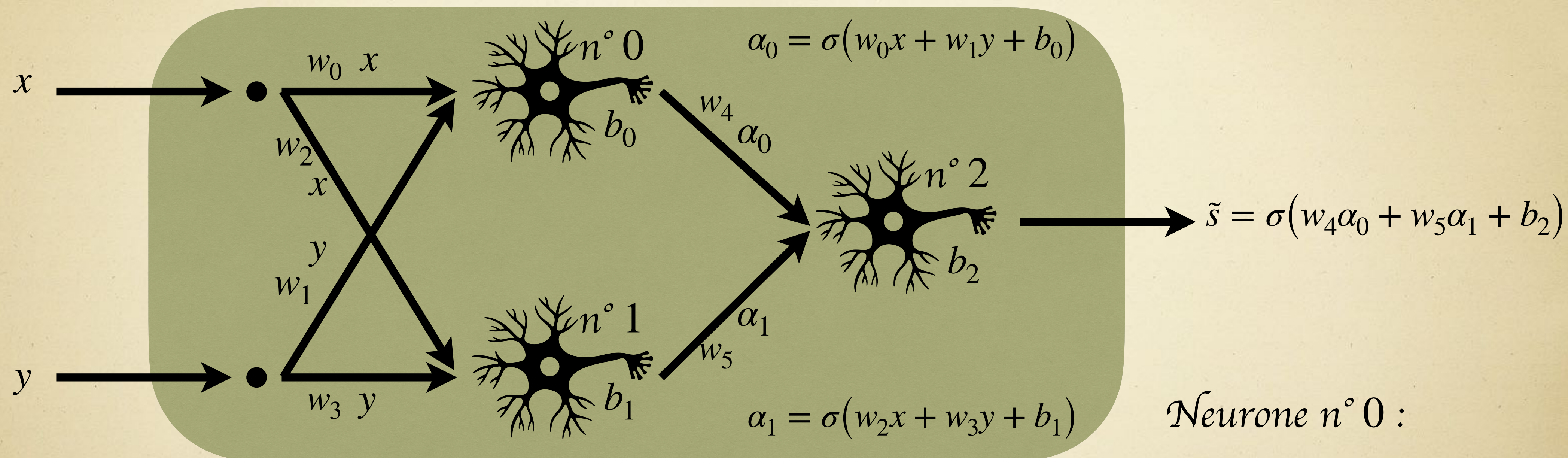
- valeurs initiales aléatoires pour  $w_i, b_i$
- propagation pour calculer  $\tilde{s}$  qui approxime  $s = x \vee y$
- calcul de l'erreur  $e = s - \tilde{s}$

Neurone n° 2 :

- $\delta_2 = e \times \sigma'(\tilde{s})$
- $w_4 = w_4 + \delta_2 \times \alpha_0$
- $w_5 = w_5 + \delta_2 \times \alpha_1$
- $b_2 = b_2 + \delta_2$

④

# TP : Perceptron affine - rétropropagation



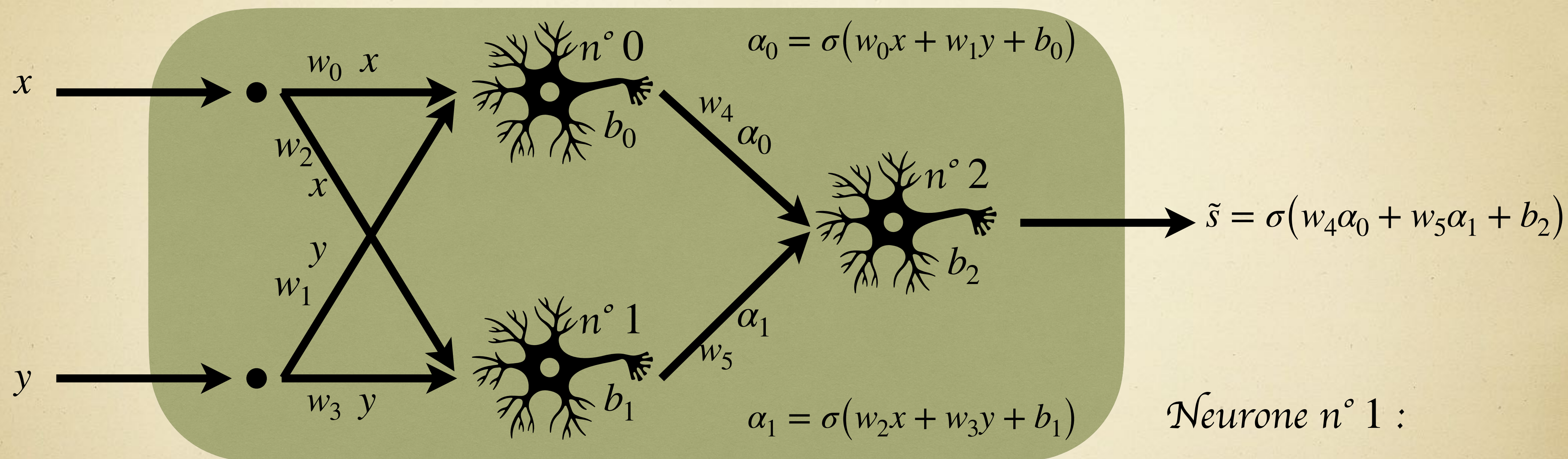
- valeurs initiales aléatoires pour  $w_i, b_i$
- propagation pour calculer  $\tilde{s}$  qui approxime  $s = x \vee y$
- calcul de l'erreur  $e = s - \tilde{s}$

Neurone  $n^{\circ} 0$  :

- $\delta_0 = \delta_2 \times w_4 \times \sigma'(\alpha_0)$
- $w_0 = w_0 + \delta_0 \times x$
- $w_1 = w_1 + \delta_0 \times y$
- $b_0 = b_0 + \delta_0$

④

# TP : Perceptron affine - rétropropagation



- valeurs initiales aléatoires pour  $w_i, b_i$
- propagation pour calculer  $\tilde{s}$  qui approxime  $s = x \vee y$
- calcul de l'erreur  $e = s - \tilde{s}$

Neurone n° 1 :

- $\delta_1 = \delta_2 \times w_5 \times \sigma'(\alpha_1)$
- $w_2 = w_2 + \delta_1 \times x$
- $w_3 = w_3 + \delta_1 \times y$
- $b_1 = b_1 + \delta_1$

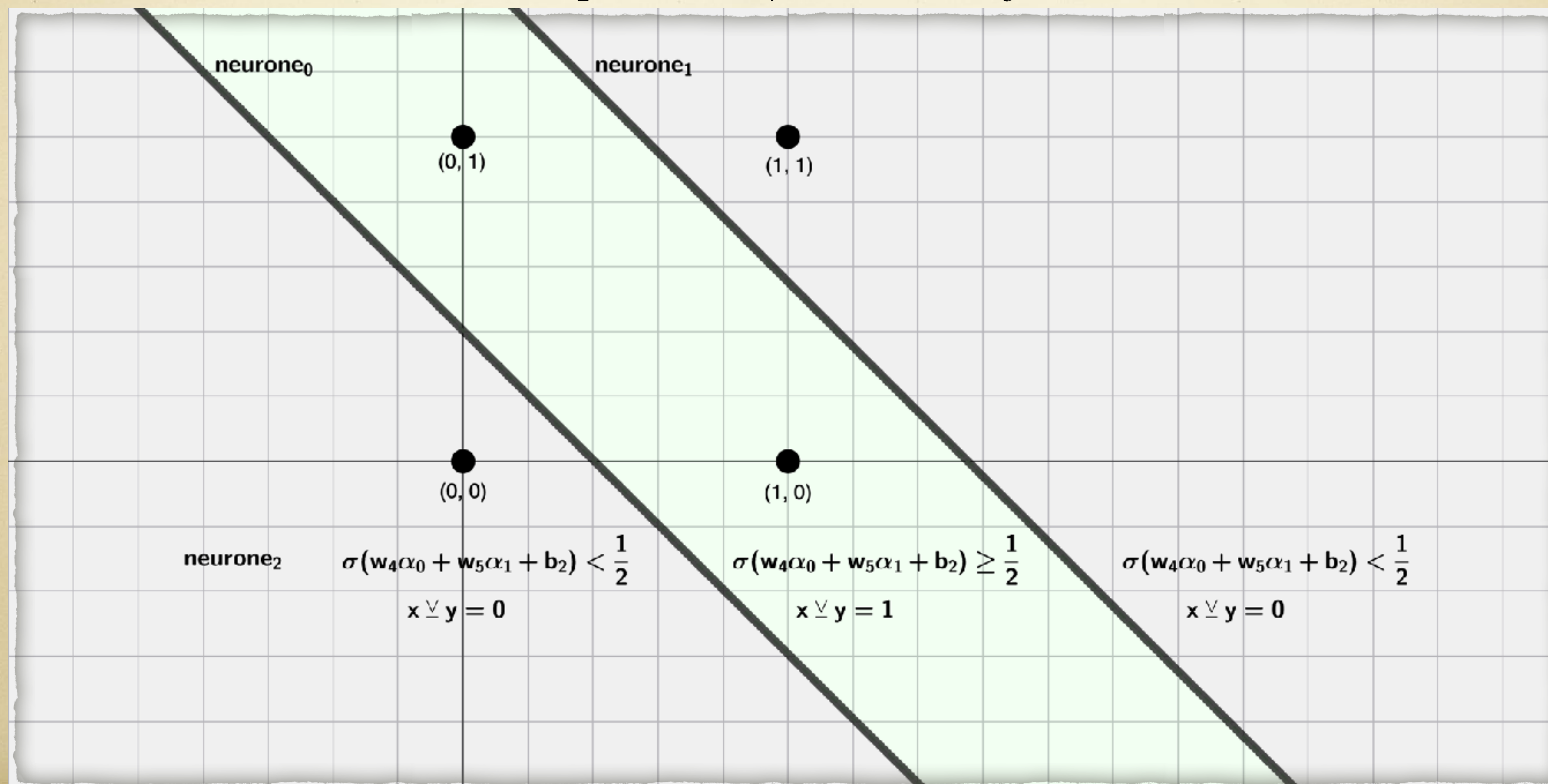
4

# TP : Perceptron affine - exemple de résultat

Neurone n° 0 :  $b_0 = 2.829, w_0 = -6.996, w_1 = -7.023$

Neurone n° 1 :  $b_1 = -7.582, w_2 = 4.881, w_3 = 4.885$

Neurone n° 2 :  $b_2 = 5.207, w_4 = -10.292, w_5 = -10.721$



# Bibliographie

*Attention is all you need*

*Language models are few-shot learners*

*Training language models to follow instructions with human feedback*

*Direct Preference Optimization: Your Language Model is Secretly a Reward Model*

*Word2Vec : Efficient estimation of Word Représentations in Vector Space*

*Démystifier l'IA*

*The Perceptron : A probabilistic model for information storage and organisation in the brain.*

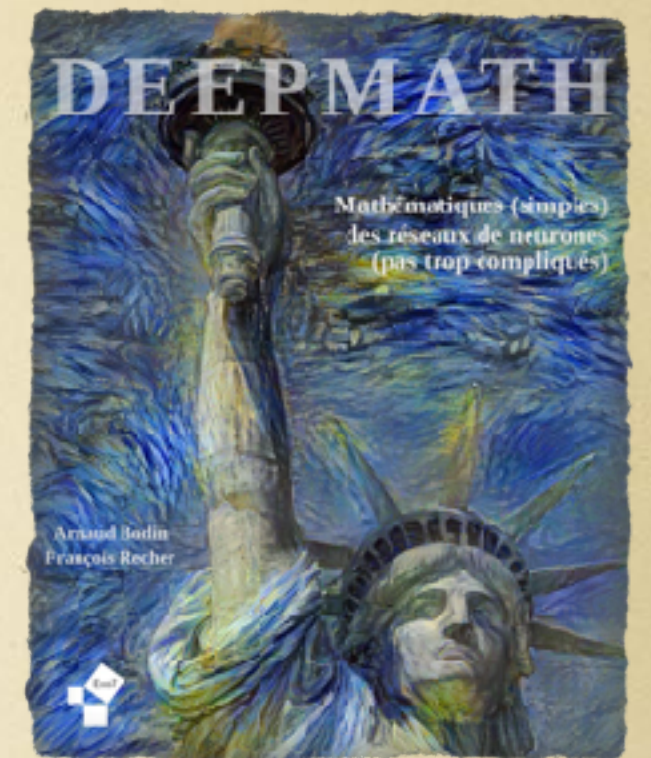


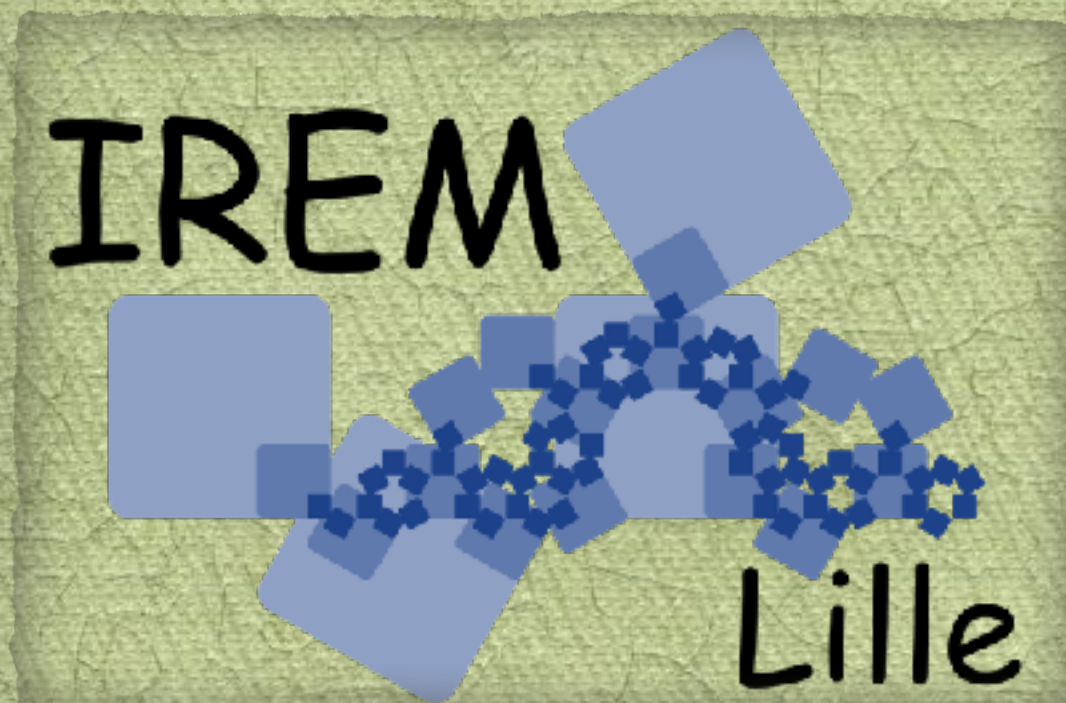
*David Louapre*



*Lê Nguyễn Hoàng*

*DeepMaths*





*Groupe Informatique*

*IREM de Lille*

*Asli Grimaud*



IA et Apprentissage Automatique

**MERCI**

Avril 2026

asli.grimaud@ac-lille.fr