Généralités sur les Large Language Models (LLM)

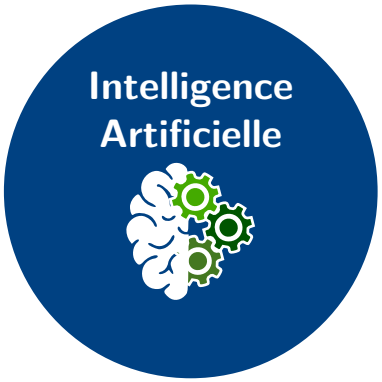
Romain LELONG

Département de Santé Numérique (DéSaN), CHU de Rouen

2024-02-20

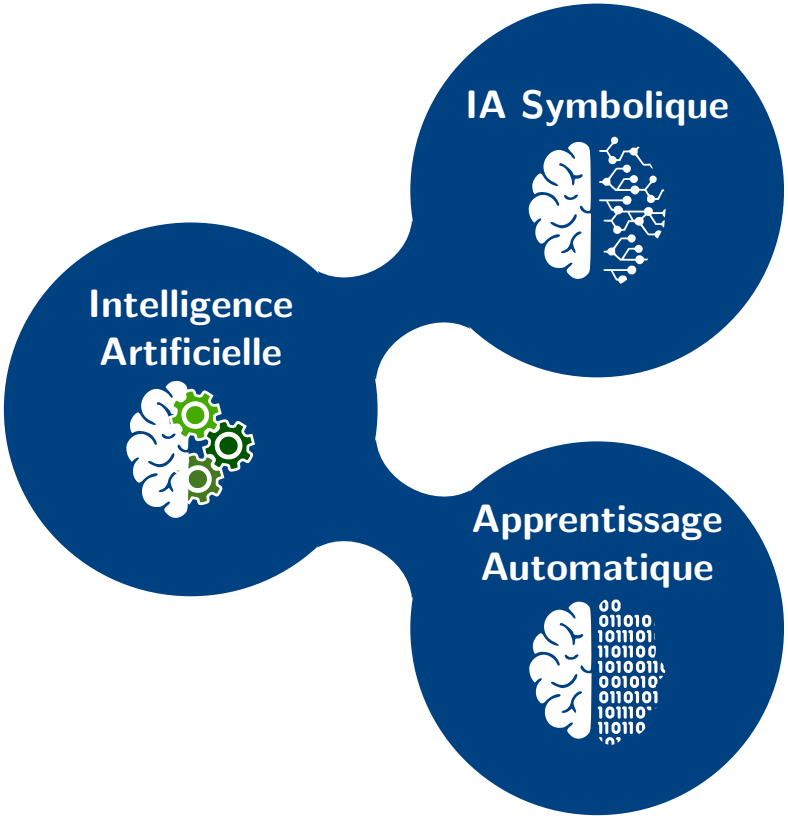
Table des matières

Le paysage de l'**I**ntelligence **A**rtificielle



Imiter les capacités cognitives humaines

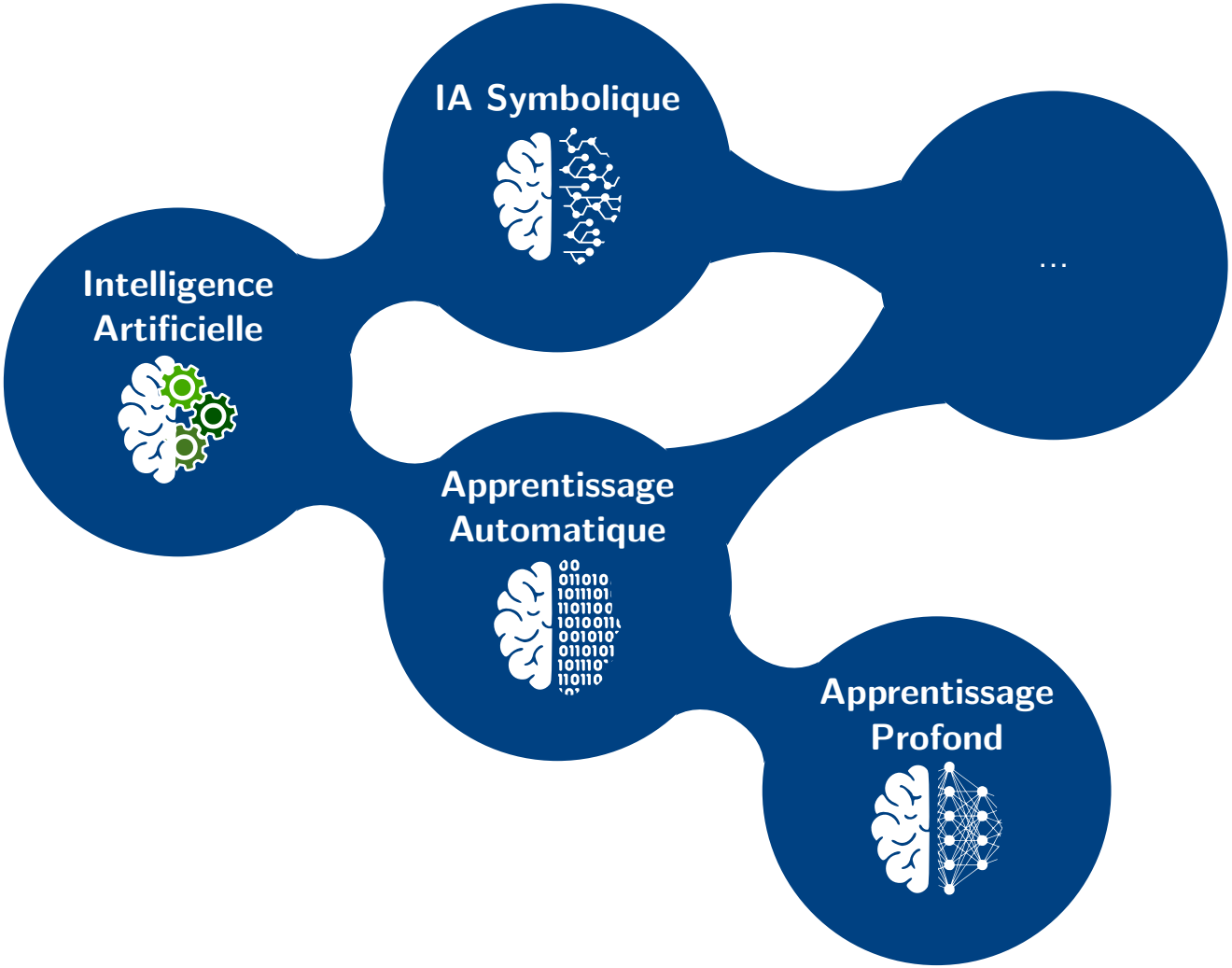
Le paysage de l'**I**ntelligence **A**rtificielle



Représenter explicitement les connaissances et utiliser de règles formelles

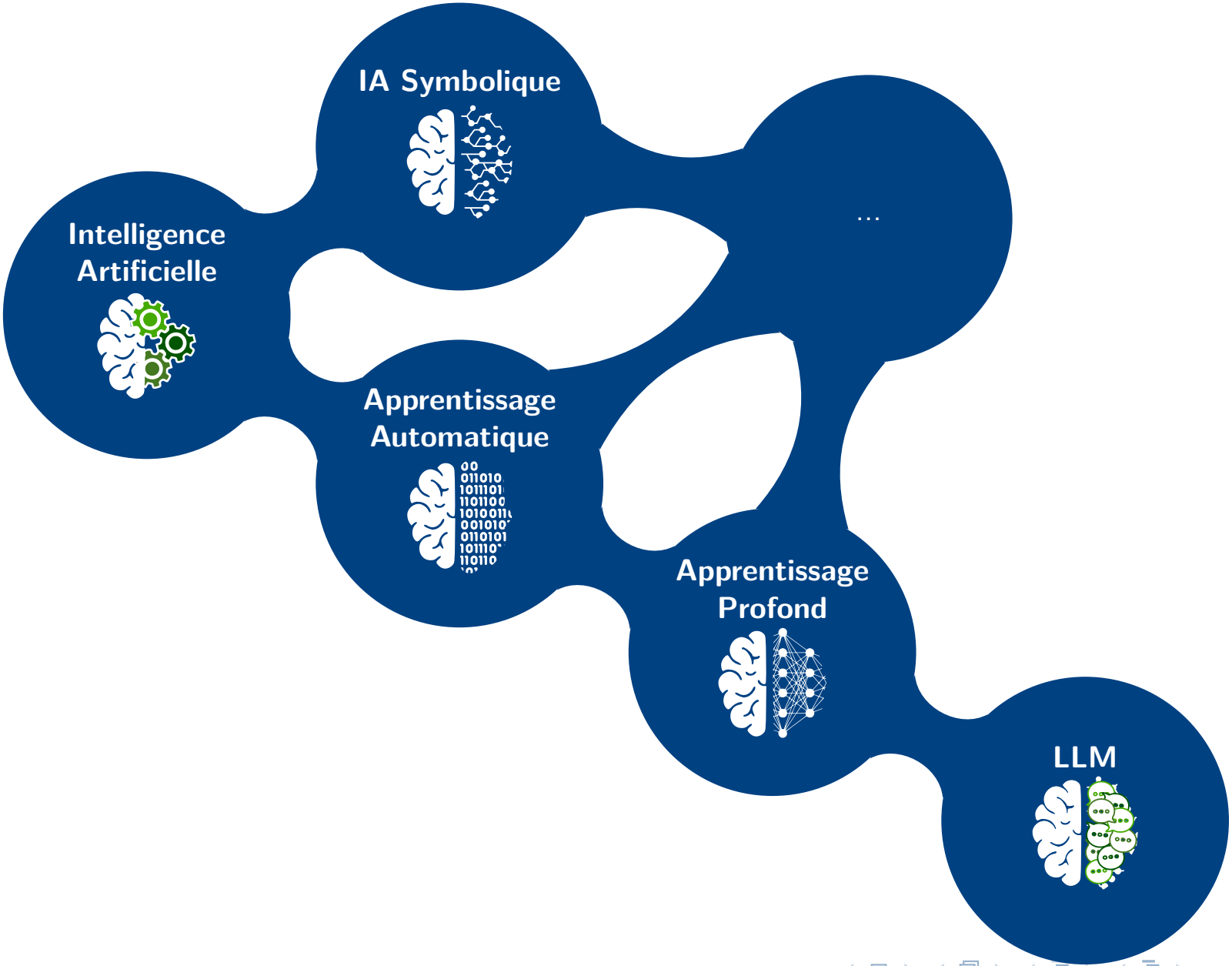
Identifier des motifs à partir des données

Le paysage de l'**I**ntelligence **A**rtificielle



Réseau de neurones multi-couches = Apprendre des représentation hiérarchique des données

Le paysage de l'**I**ntelligence **A**rtificielle



Qu'est ce qu'un LLM ?

Définition (**L**arge **L**anguage **M**odel (LLM))

Ils constituent une classe particulière de modèles d'**I**ntelligence **A**rtificielle (IA) se distinguant notamment par :

Son objectifs : *Interpréter*, *comprendre*, *représenter* et même *générer* du texte en langage naturel et plus généralement effectuer des tâches de **T**raitement **A**utomatique du **L**angage **N**aturel (TALN).

Son architecture : *réseaux de neurones profonds* avec différentes types architectures « historiques » (e.g. **R**éseaux de **N**eurones **C**onvolutionnels (CNN), **R**éseaux de **N**eurones **R**écurrents (RNN), etc.) et une architecture dominante aujourd'hui : *Transformers*.

Son pré-entraînement : sur un corpus de données *textuelles non annotées* (i.e. non-supervisé) et *massif* .

Son Fine-Tunning : *sur-entraînement* les rendant adaptable et flexible aux différent cas d'usage.

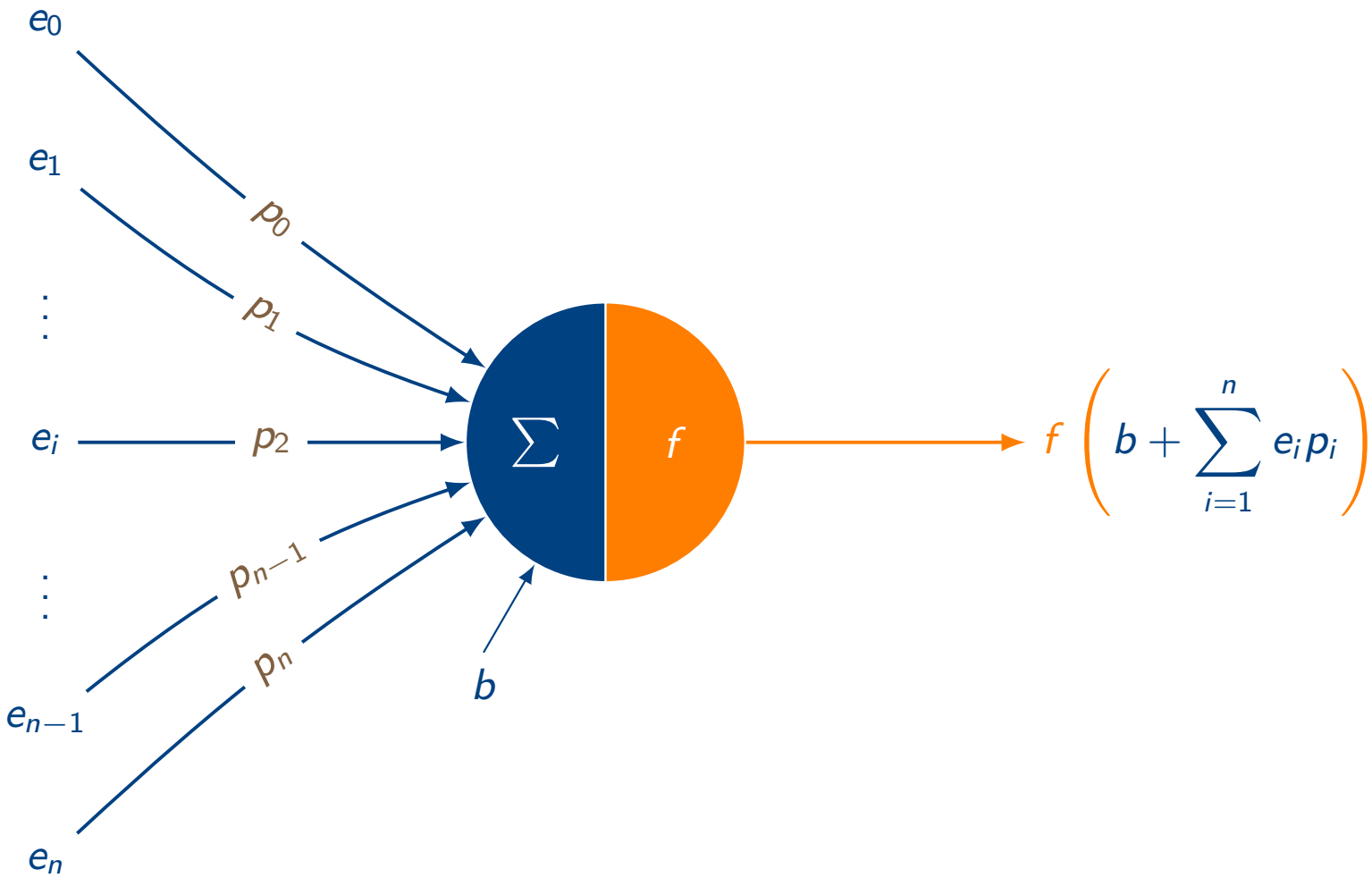
Ses capacité multimodales (MLLM) : génération de texte, d'image, de son, etc.

Apprentissage vs. Fine-Tunning

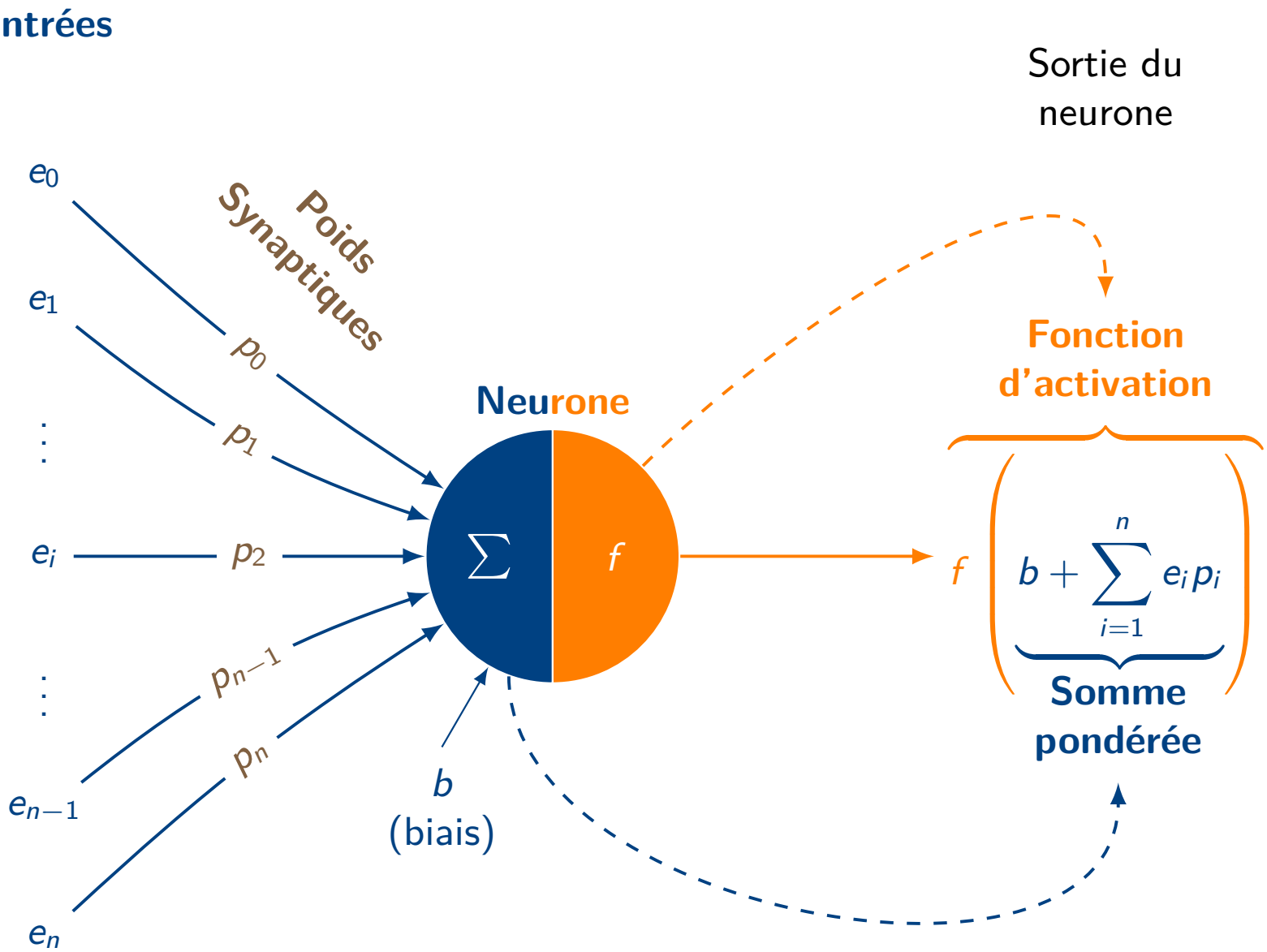
	Pré-Entraînement	Fine-Tunning
Apprentissage	*(Non)-Supervisé* Corpus non étiqueté (e.g. Pages Web, de livres, articles, etc.)	*Supervisé* Corpus étiqueté et dépendant de l'objectif du Fine-Tunning.
Objectif	*Compréhension du langage* Apprendre au modèle à comprendre et représenter des mots, des phrases et à capturer les informations sémantiques et syntaxiques du langage.	*Spécialisation et adaptation* Lui apprendre à réaliser des tâche particulières ou spécifiques (e.g. classification de texte, traduction automatique, la génération de texte, domaine médical etc.)
Ajustement paramètres	*Globale* Ajustement de l'ensemble des paramètres du réseaux de neurone (e.g. 175 milliards pour GPT–3)	*Ciblé* Ajustement de couches supérieures (implémentations spécifiques relatif aux objectifs).
Taille corpus	*Massif* Plusieurs centaines de Go (e.g. 570 Go pour GPT–3), Plusieurs centaines de milliards de mots	*Très variable* Dépend des objectifs et de la complexité des tâches visées (e.g. centaines de milliers de notes cliniques annotées pour spécialiser un LLM au domaine médical)

Table des matières

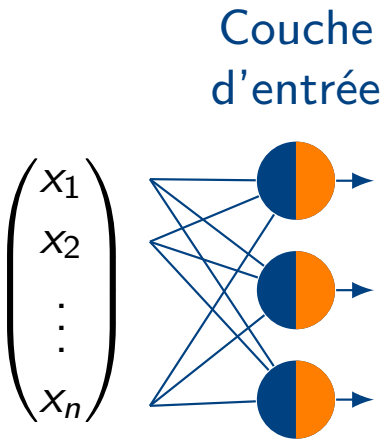
La brique élémentaire de l'apprentissage profond



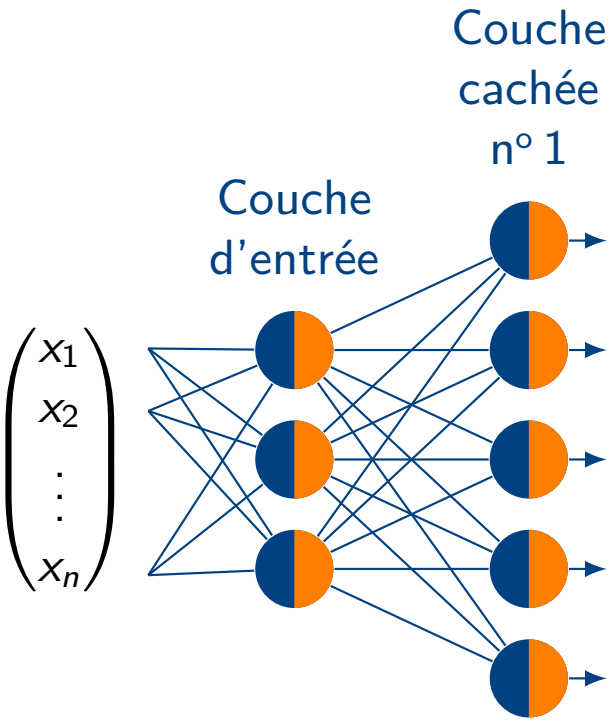
La brique élémentaire de l'apprentissage profond



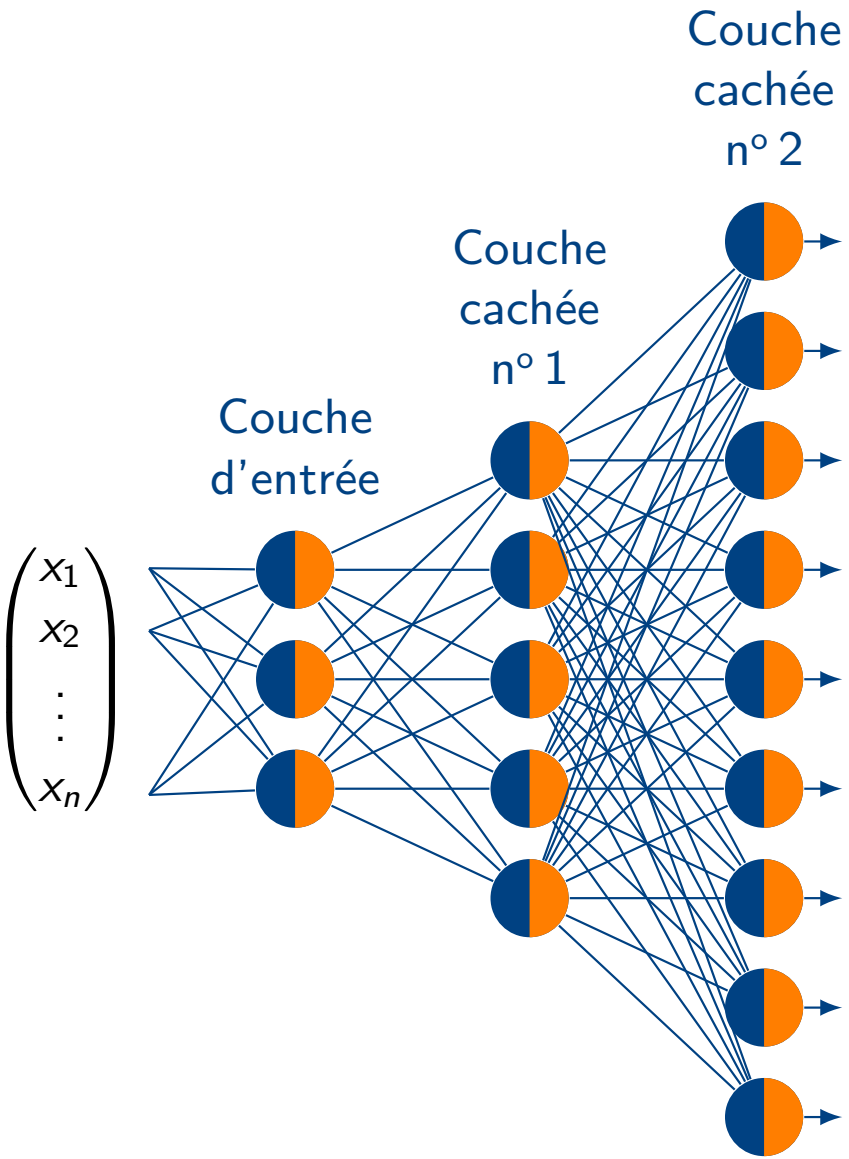
Qu'est ce qu'un réseaux de neurones



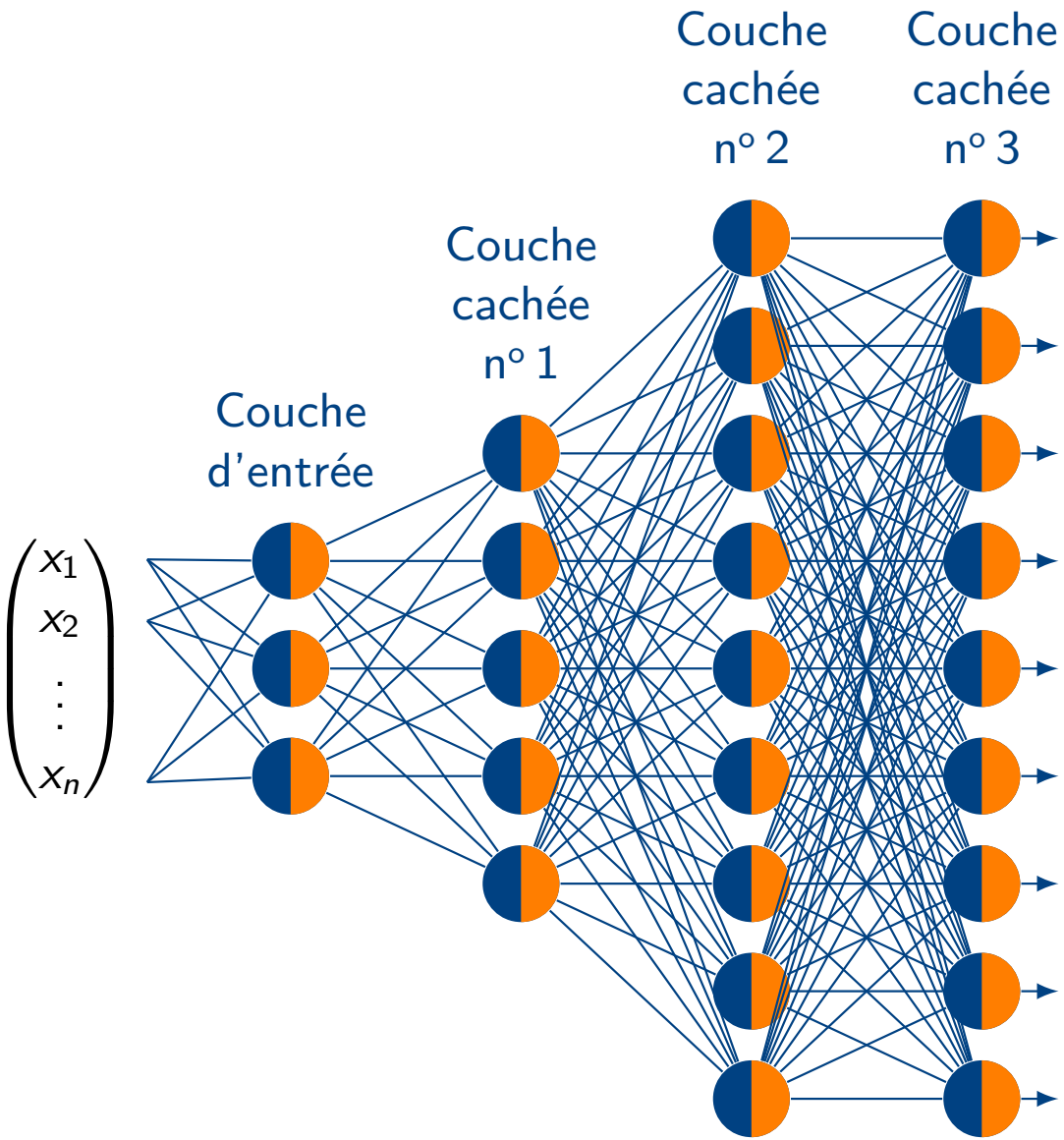
Qu'est ce qu'un réseaux de neurones



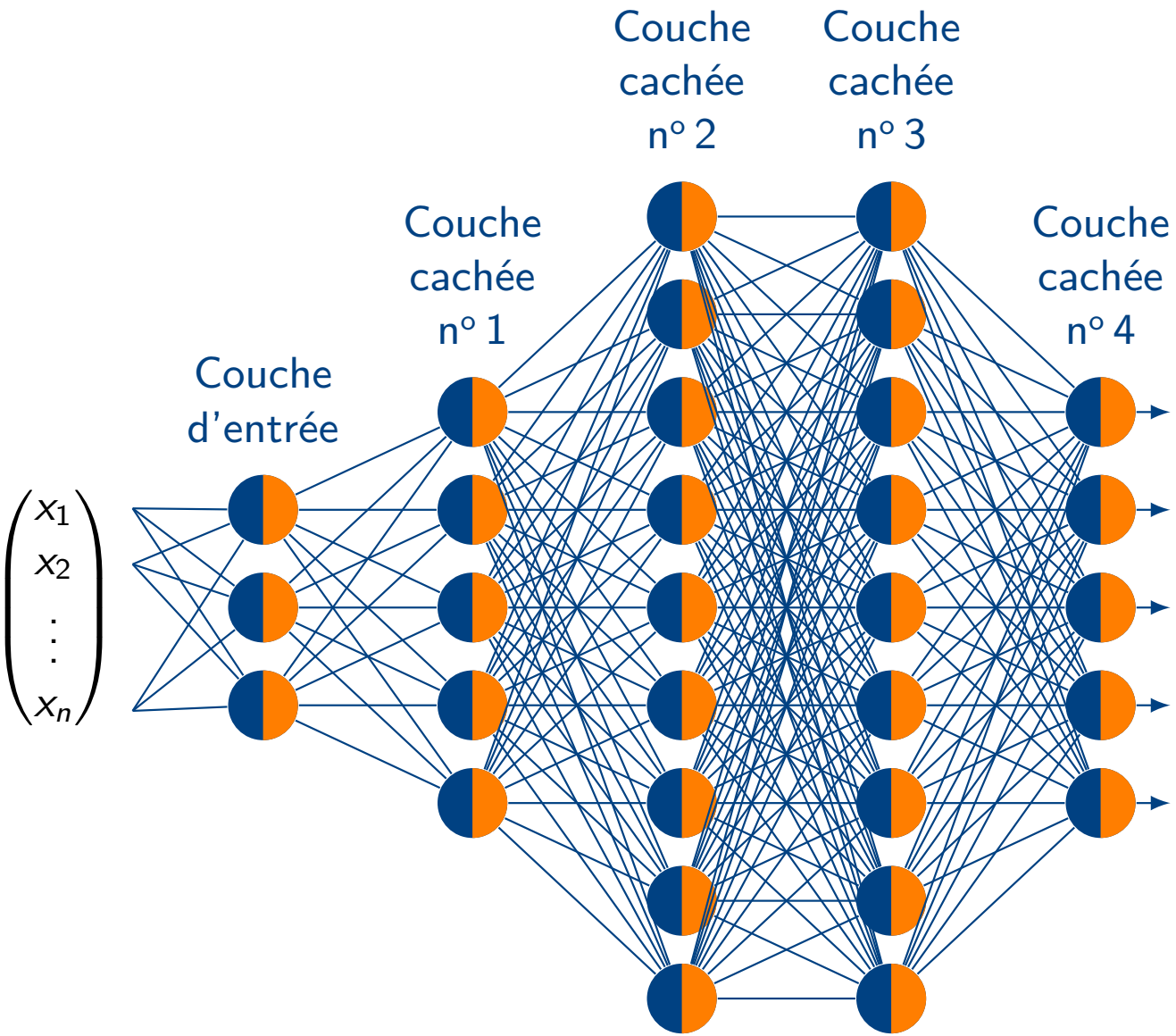
Qu'est ce qu'un réseaux de neurones



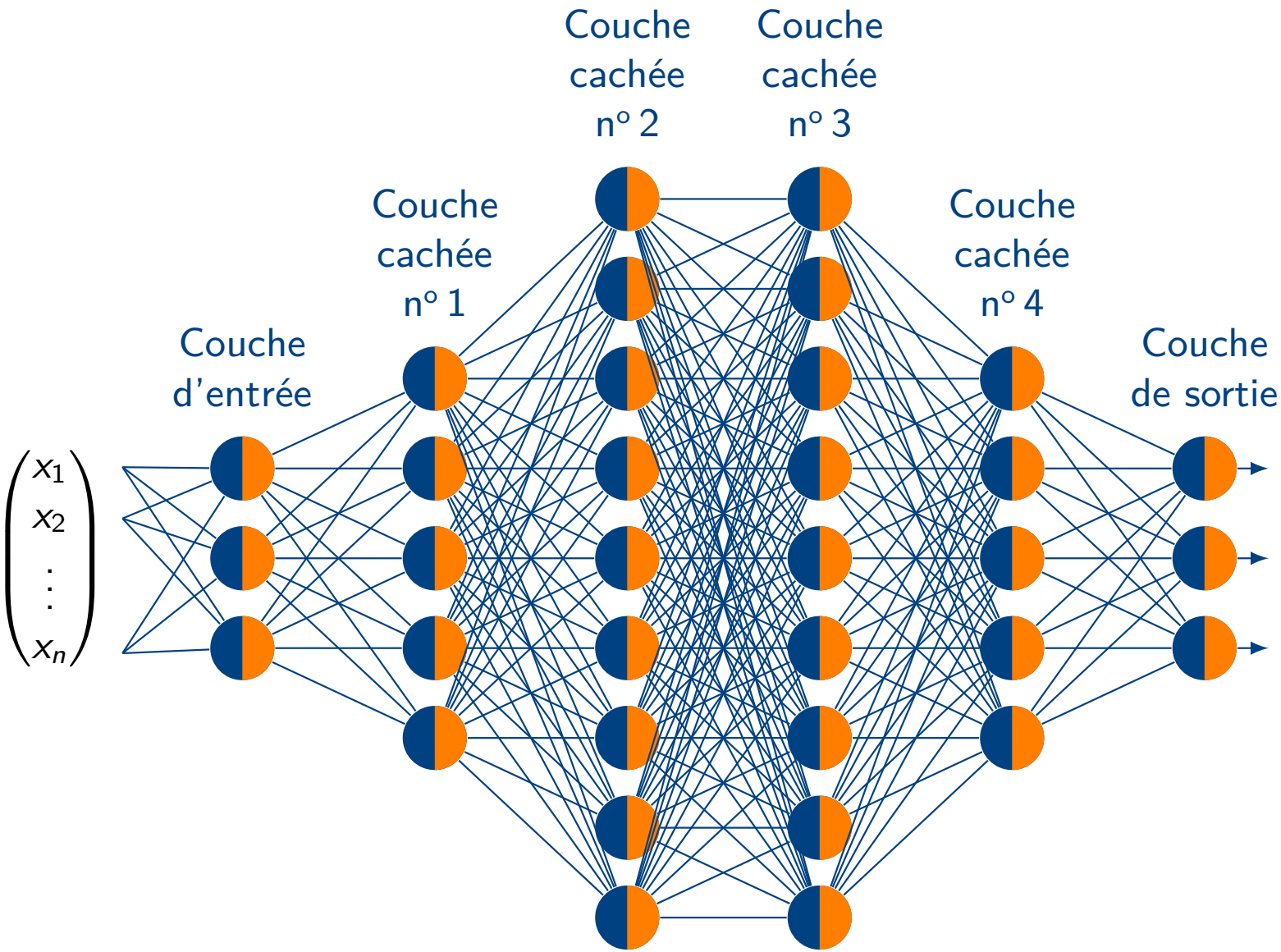
Qu'est ce qu'un réseaux de neurones



Qu'est ce qu'un réseaux de neurones



Qu'est ce qu'un réseaux de neurones



Qu'est ce qu'un réseaux de neurones

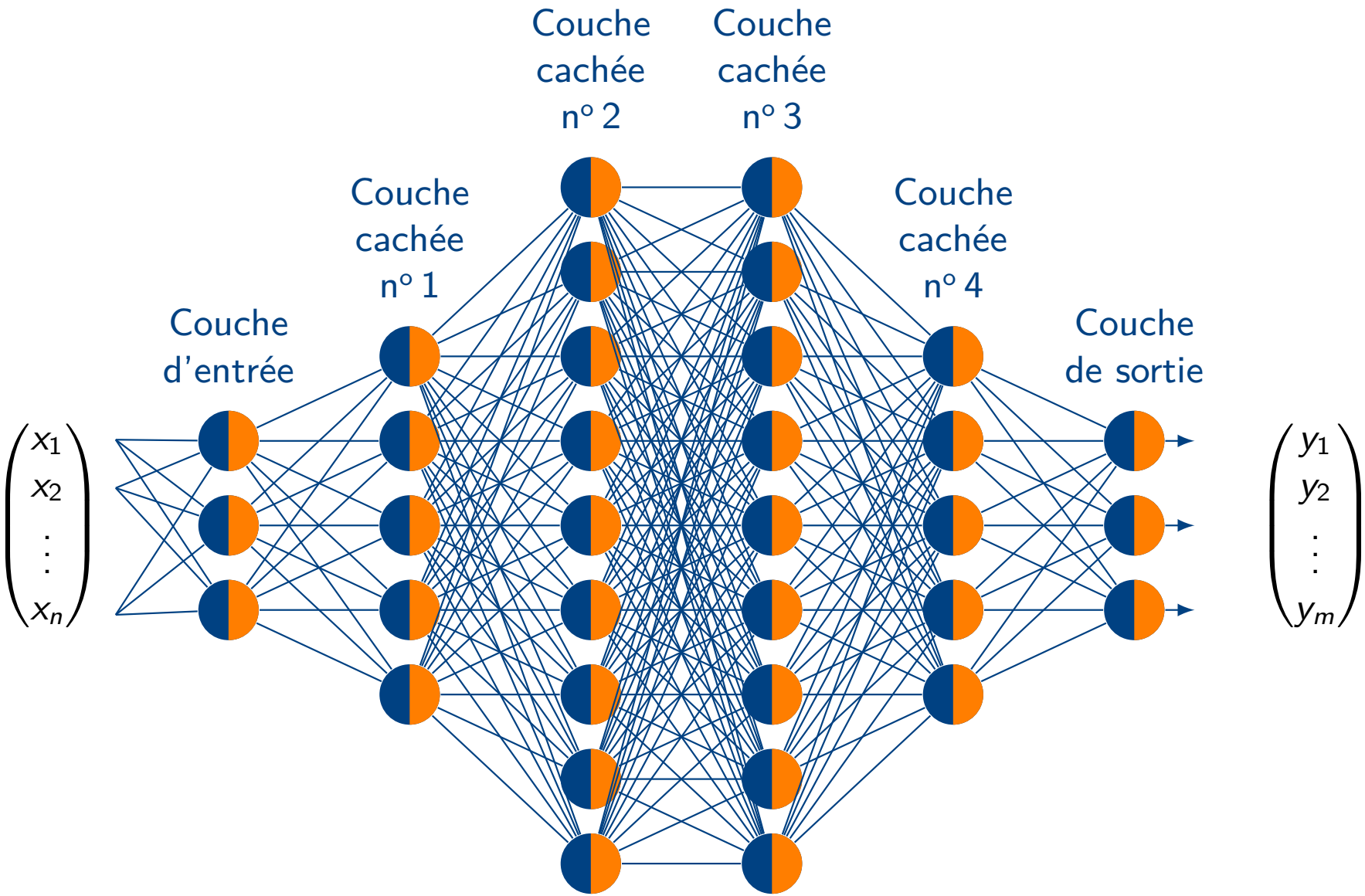
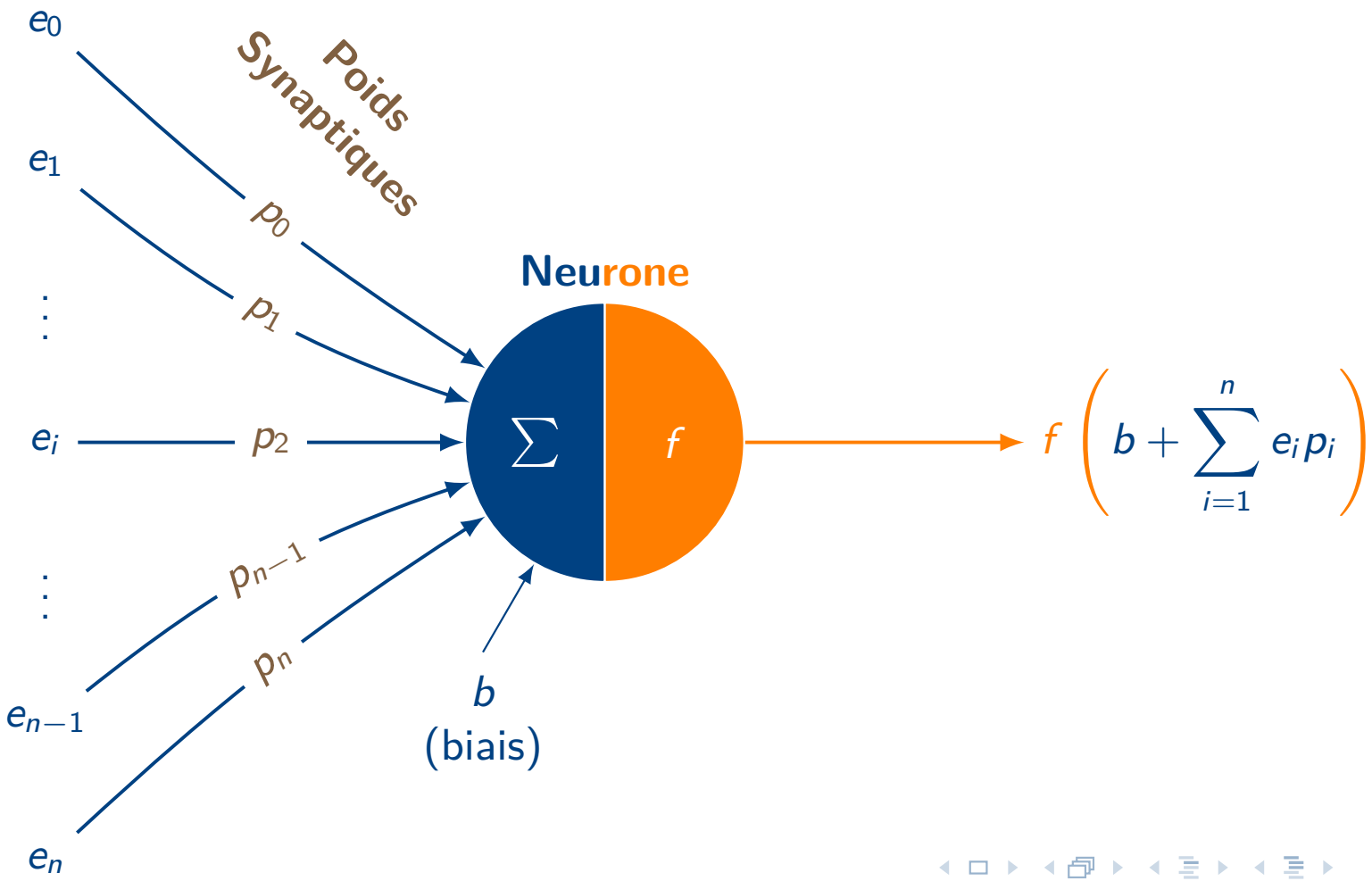


Table des matières

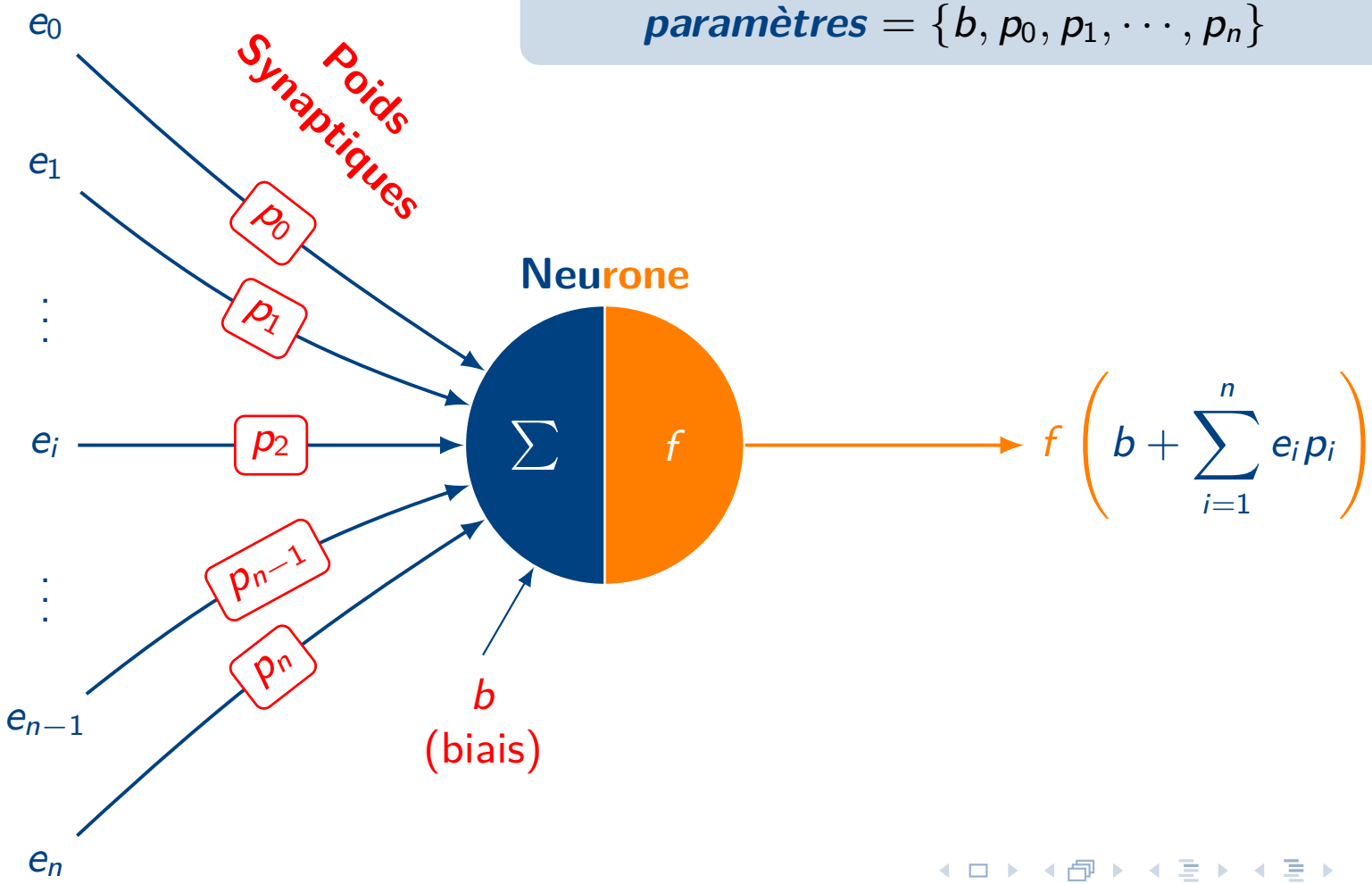
Entrées


e_0
e_1
e_i
e_{n-1}
e_n
Poids Synaptiques
p_0
p_1
p_2
p_{n-1}
p_n
Neurone
$\sum$
f
b
(biais)
$f\left(b+\sum_{i=1}^{n} e_i p_i\right)$

Entrées

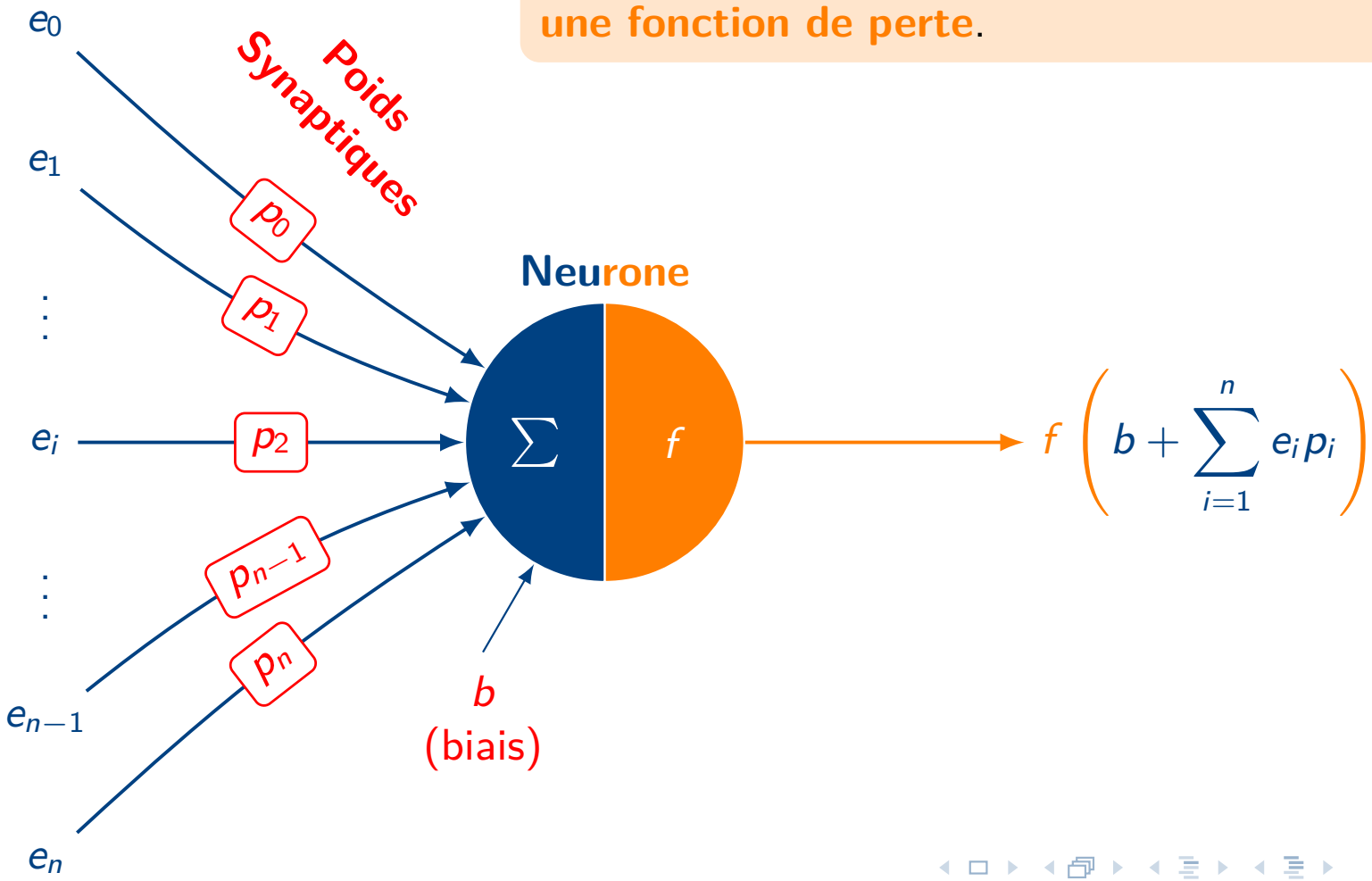
Définition (Paramètres d'un neurones)

$$\textbf{\textit{paramètres}} = \{b, p_0, p_1, \cdots, p_n\}$$

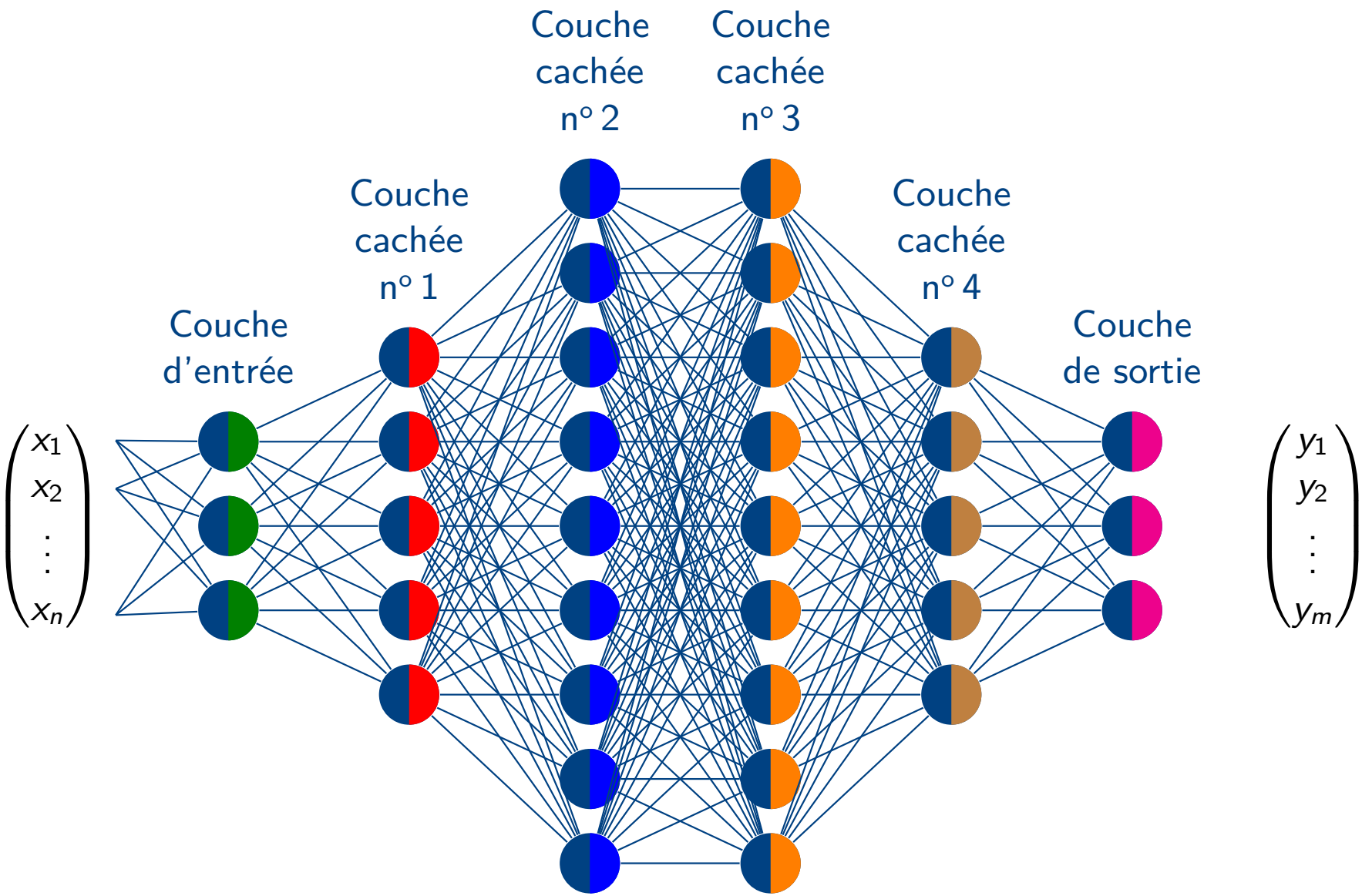


Propriété (Phase d'entrainement initiale)

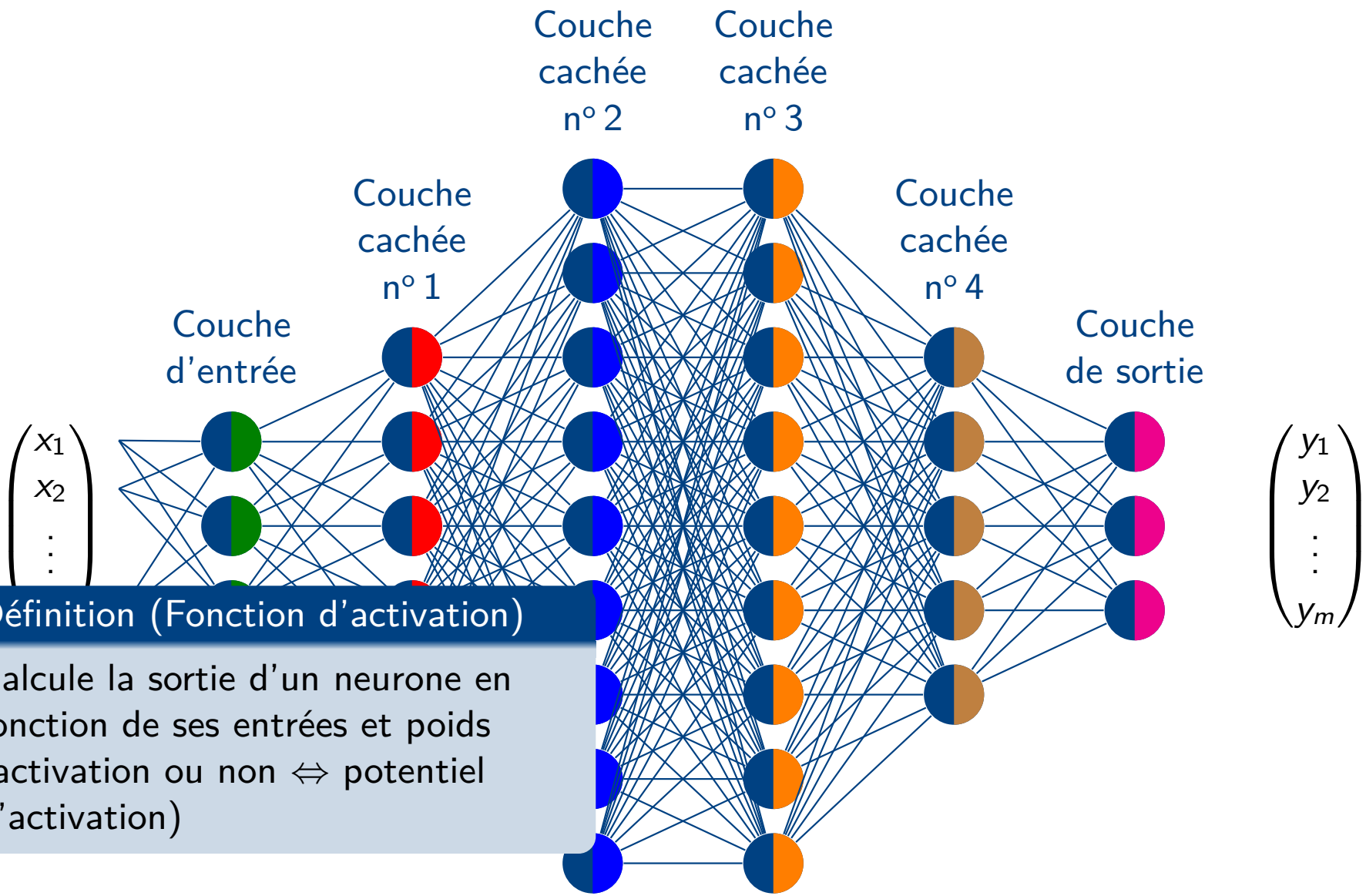
Consiste à trouver les valeurs de paramètre des neurones (i.e. poids synaptiques + biais) optimaux. C'est à dire qui **minimise une fonction de perte**.



La fonction d'activation

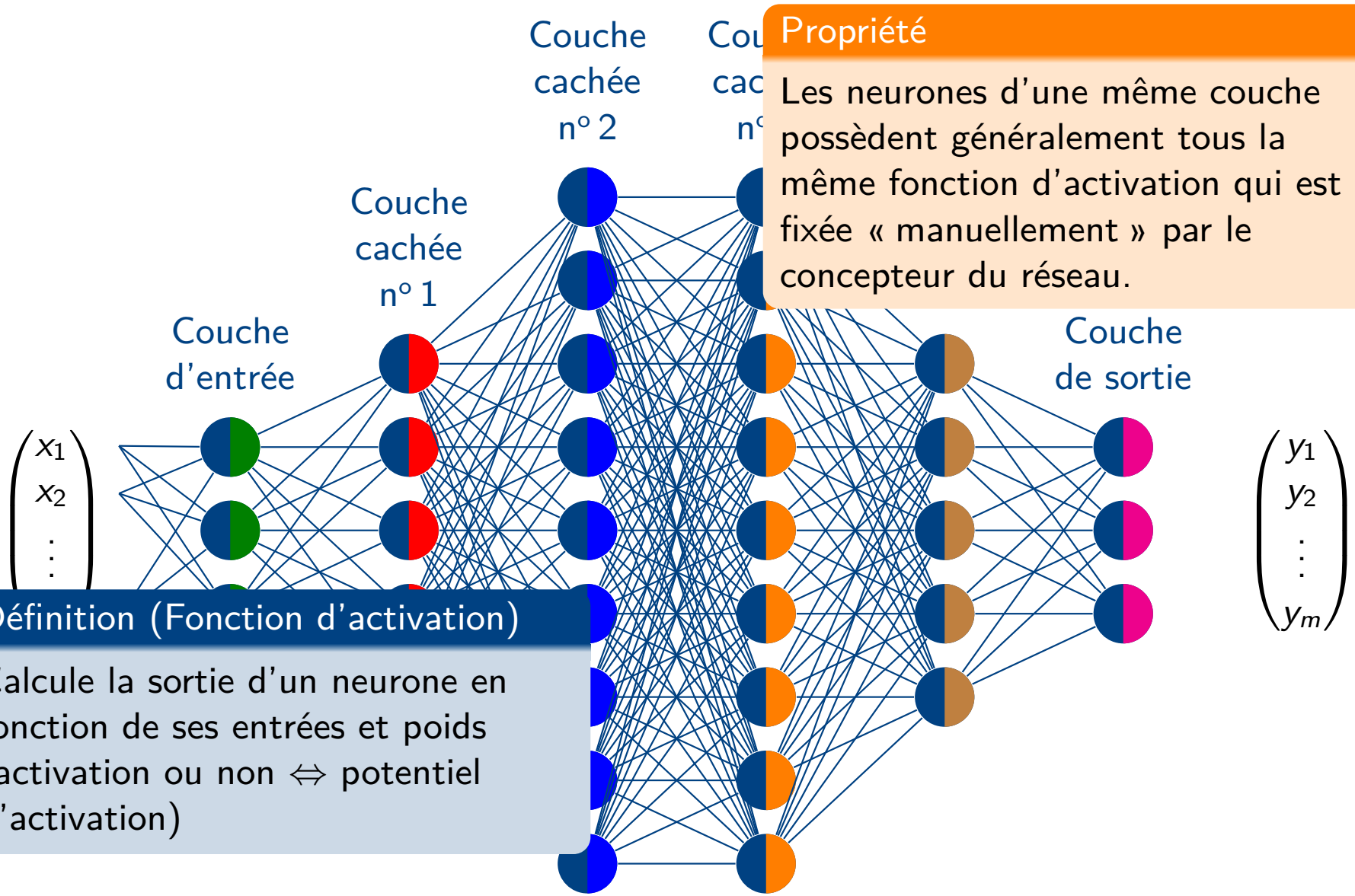


La fonction d'activation



Définition (Fonction d'activation)

Calcule la sortie d'un neurone en fonction de ses entrées et poids (activation ou non $\Leftrightarrow$ potentiel d'activation)



Définition (Fonction d'activation)

Calcule la sortie d'un neurone en fonction de ses entrées et poids (activation ou non $\Leftrightarrow$ potentiel d'activation)

Propriété

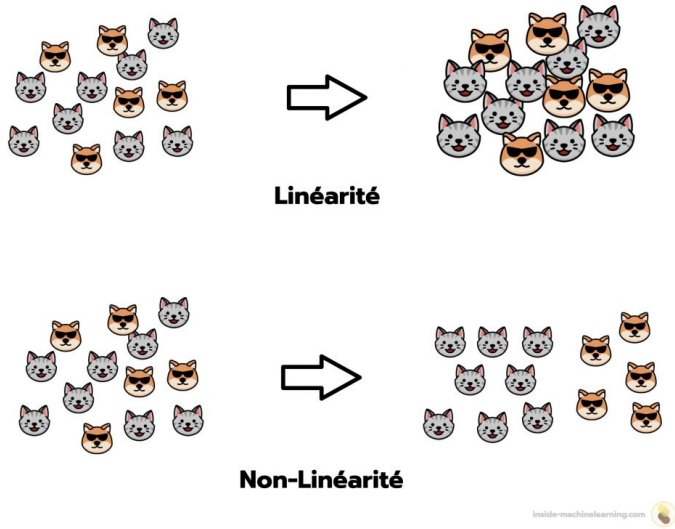
Les neurones d'une même couche possèdent généralement tous la même fonction d'activation qui est fixée « manuellement » par le concepteur du réseau.

Importance de la Non-Linéarité

Propriété (Fonction d'activation Non-Linéaire)

Dans les LLM, les fonctions d'activations utilisées sont le plus souvent **Non-Linaires**. Cette non linéarité donne la capacité ...

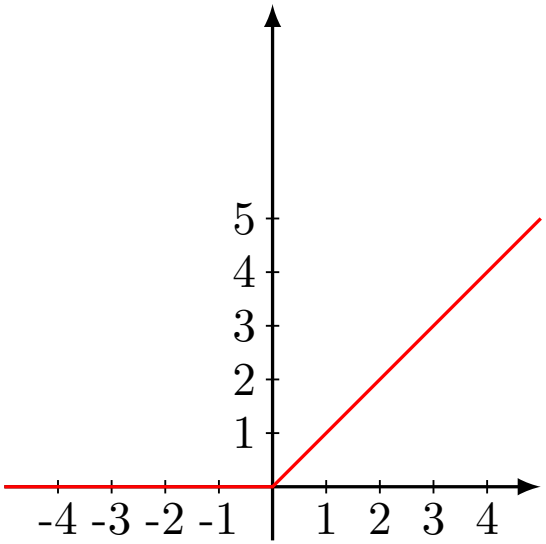
- de modéliser des représentations et des relation non linéaires des données (non linéarité du monde réel).
- d'apprendre des représentations riches/complexes des données. *A contrario*, composer *n* couches de neurones linéaires n'aboutirait qu'à une transformation linéaire globale.
- de permettre un propagation efficace du gradient lors de l'apprentissage par rétro-propagation. *A contrario*, une dérivée constante par rapport au poids du réseau ne permettrait pas un ajustement significatif de ces poids.



Exemples de Fonction d'activation

Rectified Linear Unit (ReLU)

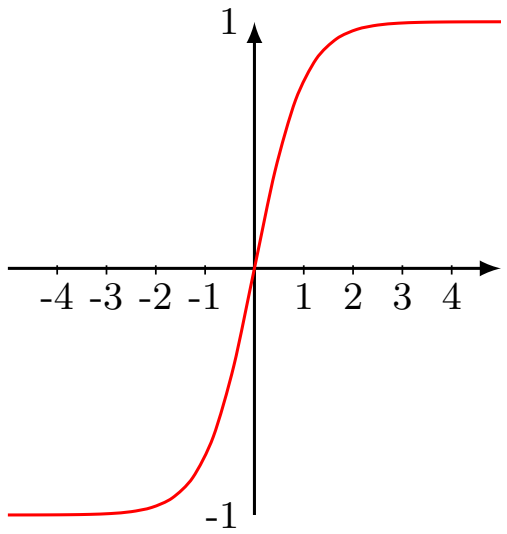
$$f(x) = max(0, x)$$



Cas d'usage : *focus sur certaines caractéristiques des données au détriment d'autres qui sont « éliminées ». Elle est utile dans les tâches de classification binaire ou éventuellement multi-classe voir de régression.*

Tangente hyperbolique

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

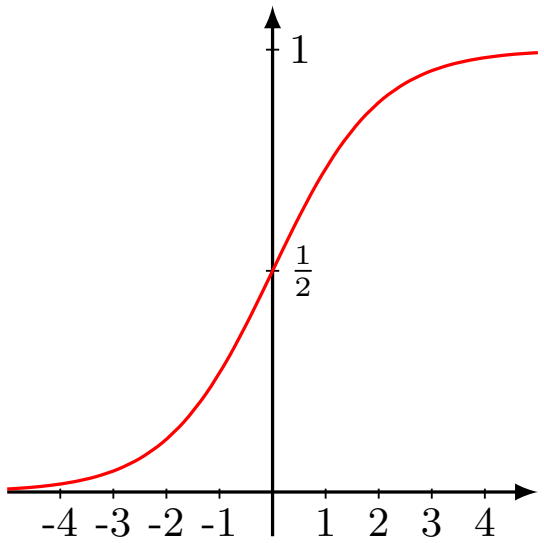


Cas d'usage : *Elle a un effet de centrage et de mise à l'échelle (≃ normalisation) des données d'entrée. Elle peut également remplacer la fonction Sigmoïde dans la dernière couche d'un modèle de classification binaire.*

Exemples de Fonction d'activation

Sigmoïde

$$f(x) = \frac{1}{1 + \exp(-x)}$$



Cas d'usage : *Interprétable comme une probabilité qu'une donnée appartiennent à une classe. Elle constitue souvent la dernière couche d'un réseaux de neurone de classification binaire*

Softmax

$$f(x_i) = \frac{e^{x_i}}{e^{x_1} + e^{x_2} + \cdots + e^{x_n}}$$

où

n le nombre total de neurones dans la couche

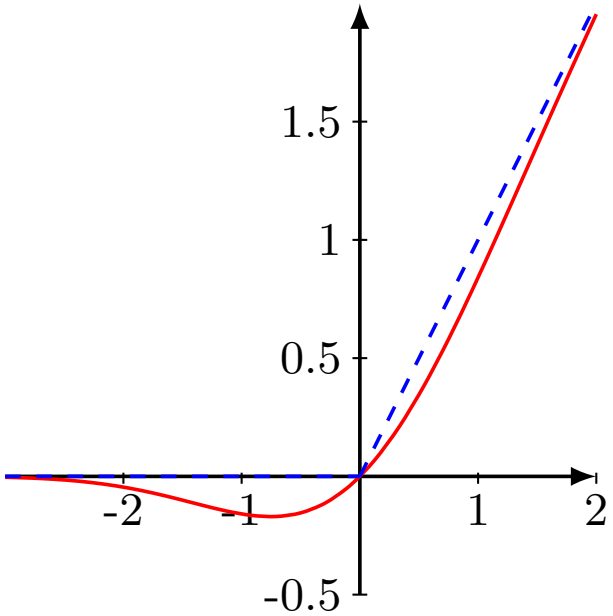
$x_i = b_i + \sum_{j=1}^{n} p_j \cdot e_j$ est le score associé au neurone i de la couche.

Cas d'usage : *Conversion des entrée/scores (logits) bruts en une distribution de probabilité sur les différentes classes. Utilisée dans la dernière couche d'un réseaux de neurones de classification multi-classe.*

Exemple de Fonction d'activation

Fonction Gaussian Error Linear Units (GELUs)

$$
\begin{aligned}
f(x) &= xP(X \leqslant x) \text{ avec } X \sim \mathcal{N}(0,1) \\
&= x\Phi(x) \text{ où } \Phi \text{ est la distribution cumulative gaussienne standard} \\
&\simeq \frac{x}{2}\left(1+\tanh\left(\sqrt{\frac{2}{\pi}}\left(x+0.044715x^3\right)\right)\right)
\end{aligned}
$$



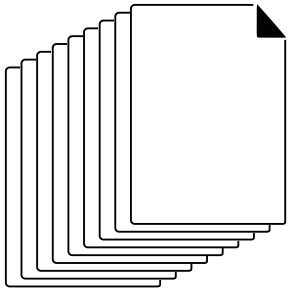
Introduite en 2016[a]

a. Hendrycks et Gimpel, "Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units".

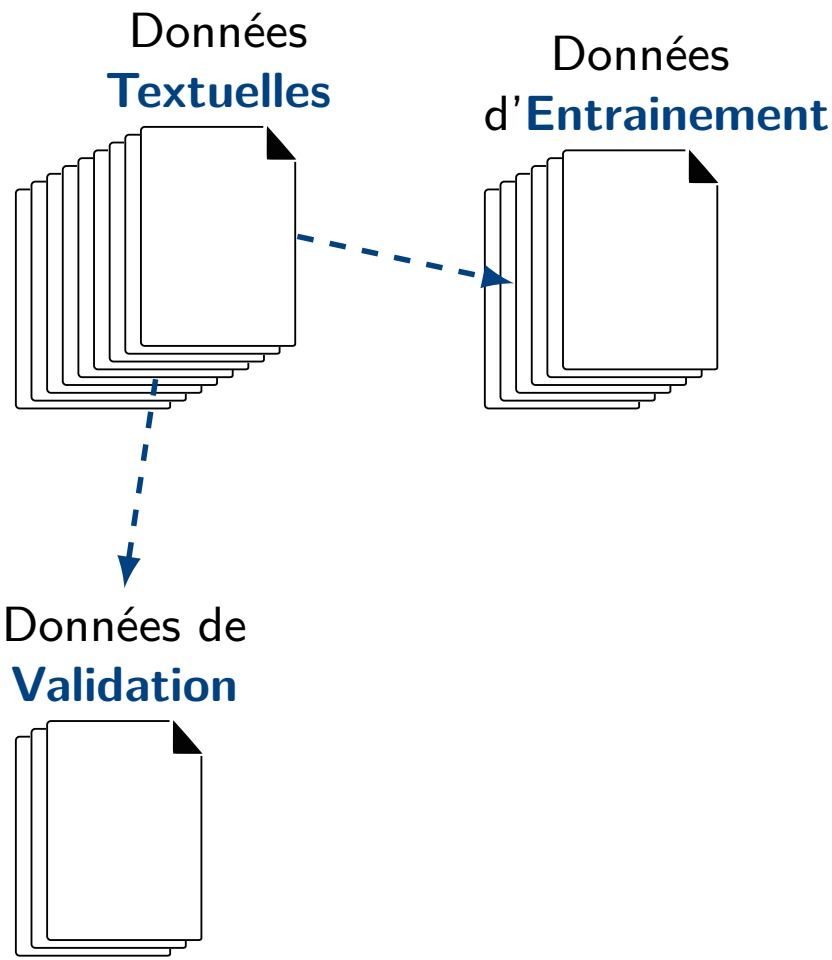
Table des matières

Phase de pré-entraînement

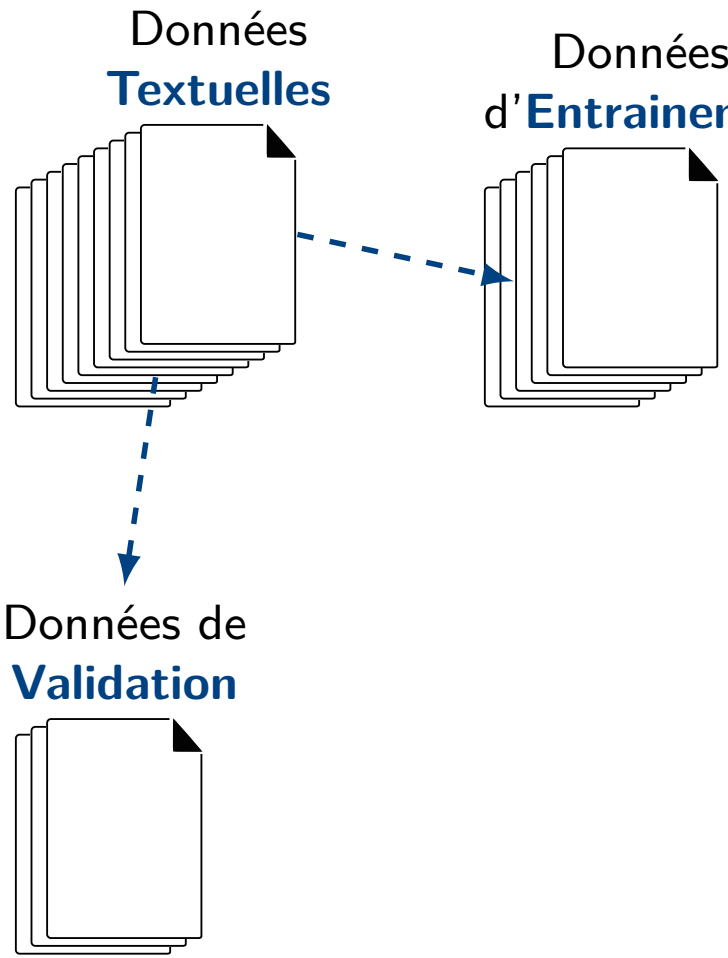
Données
Textuelles



Phase de pré-entraînement

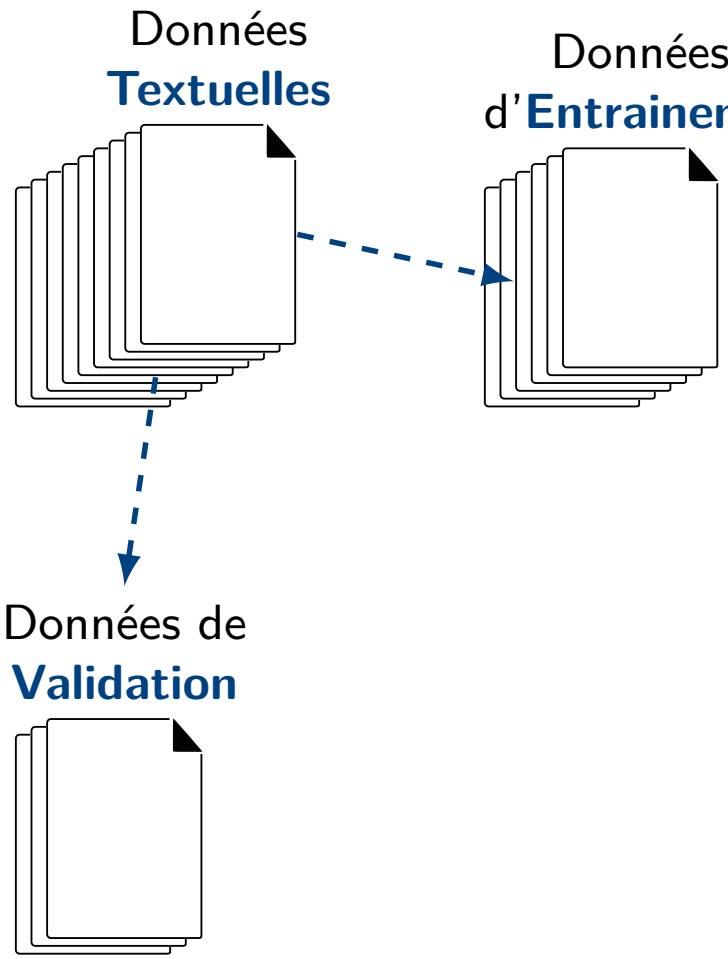


Phase de pré-entraînement



Séquence de tokens		token suivants
$[a_1, a_2, \cdots]$	$\rightarrow$	a ✓
$[b_1, b_2, \cdots]$	$\rightarrow$	b ✓
$[c_1, c_2, \cdots]$	$\rightarrow$	c ✓
$\cdots$		$\cdots$

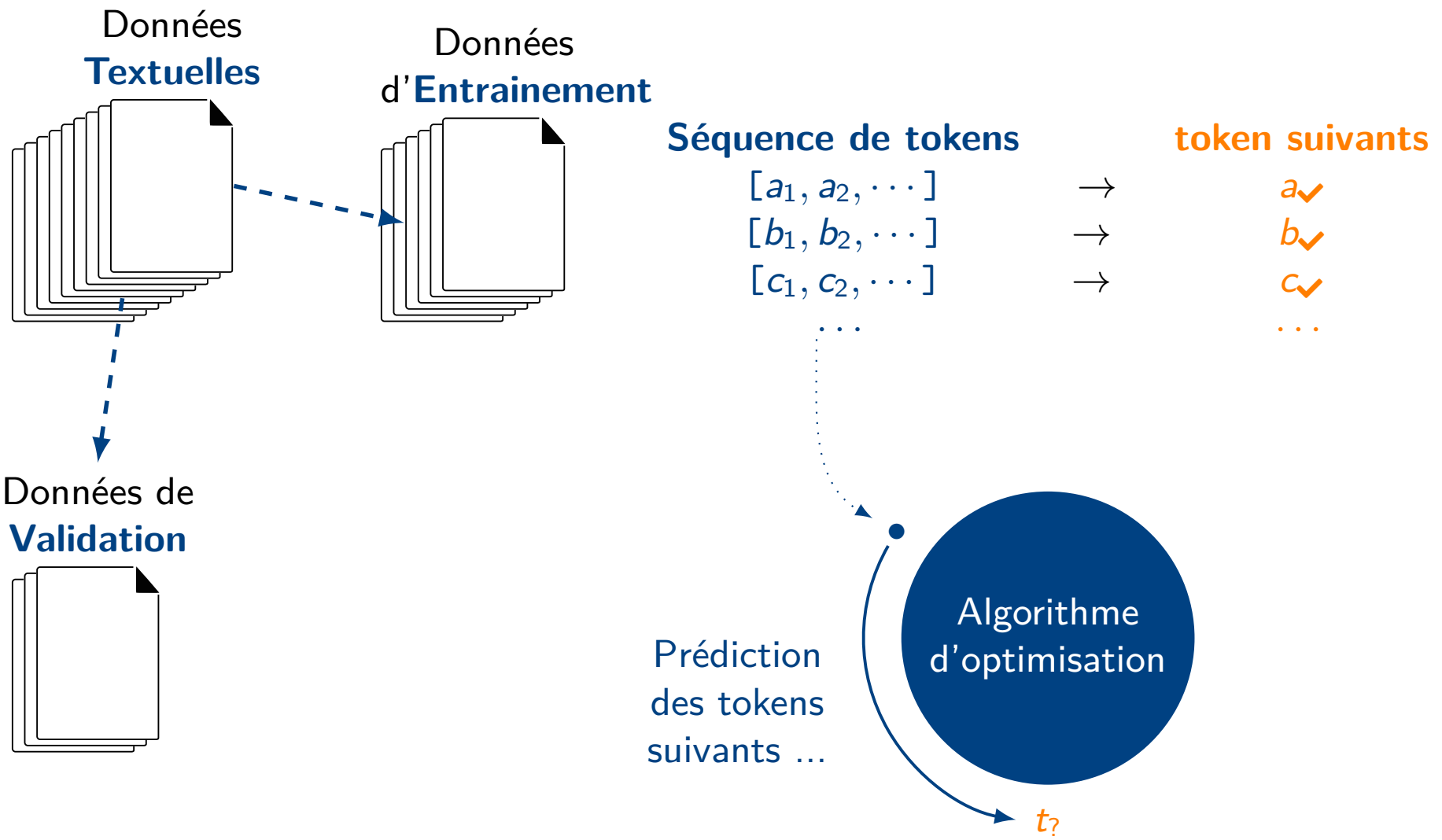
Phase de pré-entraînement



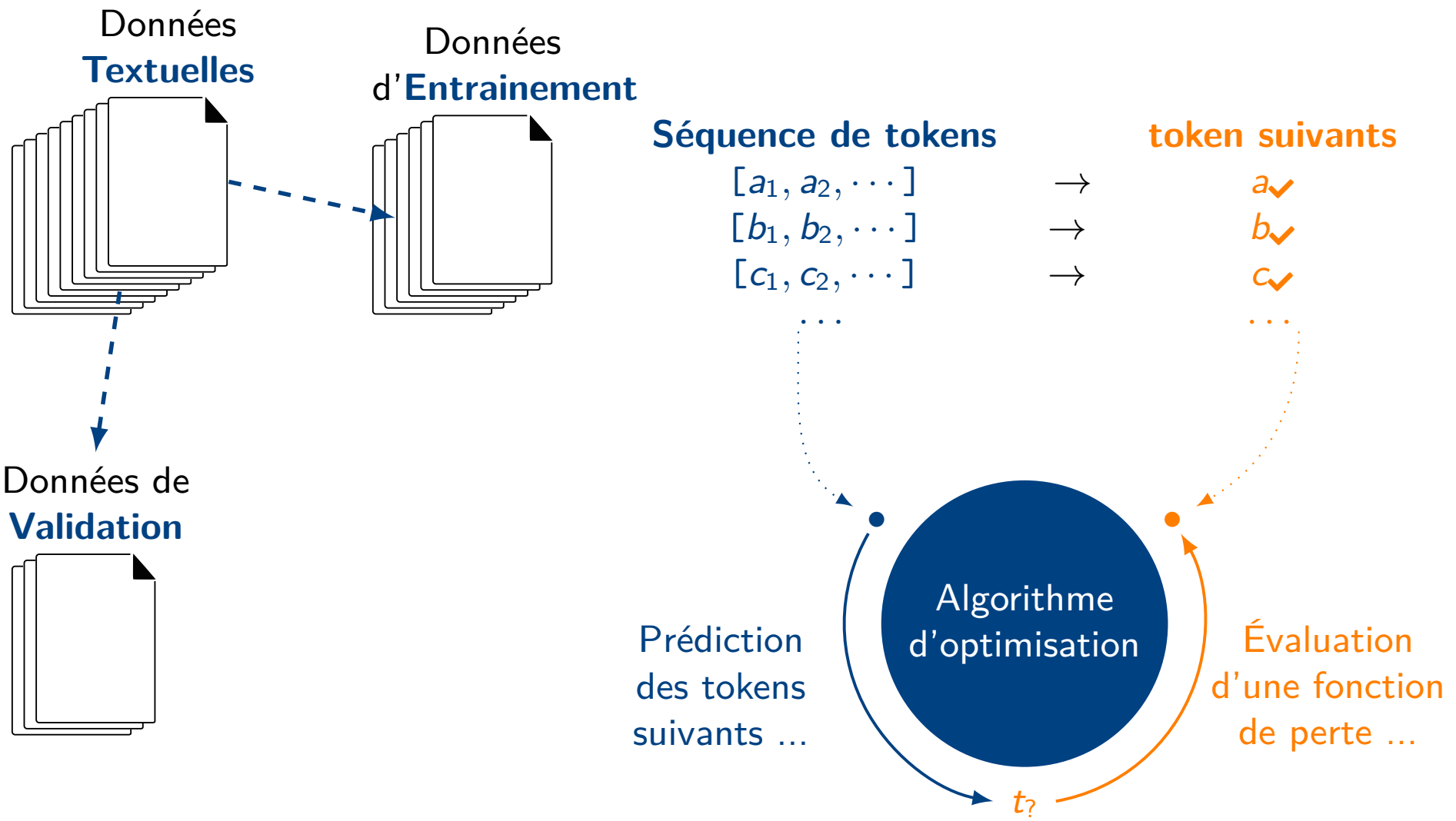
Séquence de tokens		**token suivants**
$[a_1, a_2, \cdots]$	$\rightarrow$	a ✓
$[b_1, b_2, \cdots]$	$\rightarrow$	b ✓
$[c_1, c_2, \cdots]$	$\rightarrow$	c ✓
$\cdots$		$\cdots$

Algorithme d'optimisation

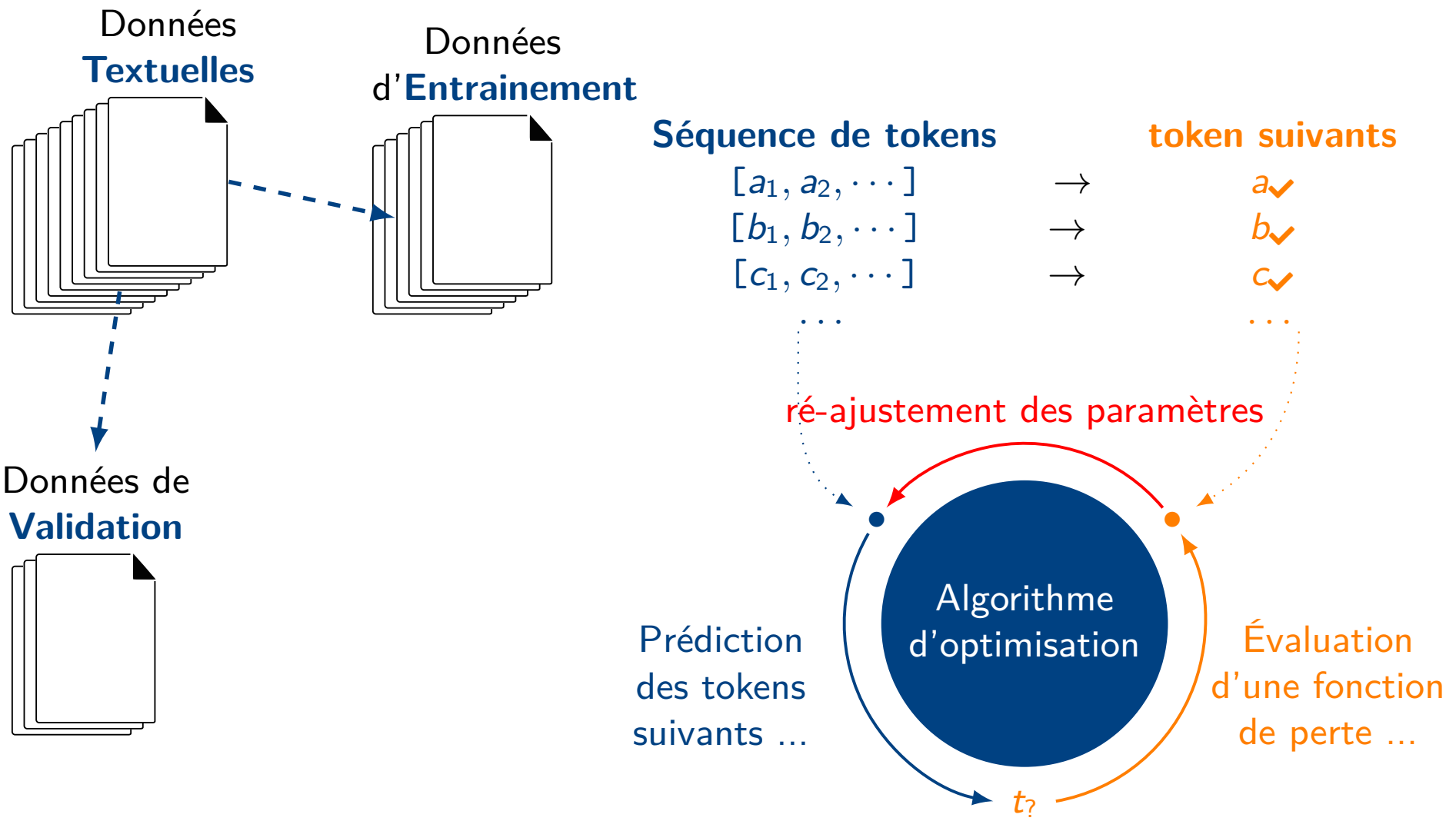
Phase de pré-entraînement



Phase de pré-entraînement



Phase de pré-entraînement



Phase de pré-entraînement

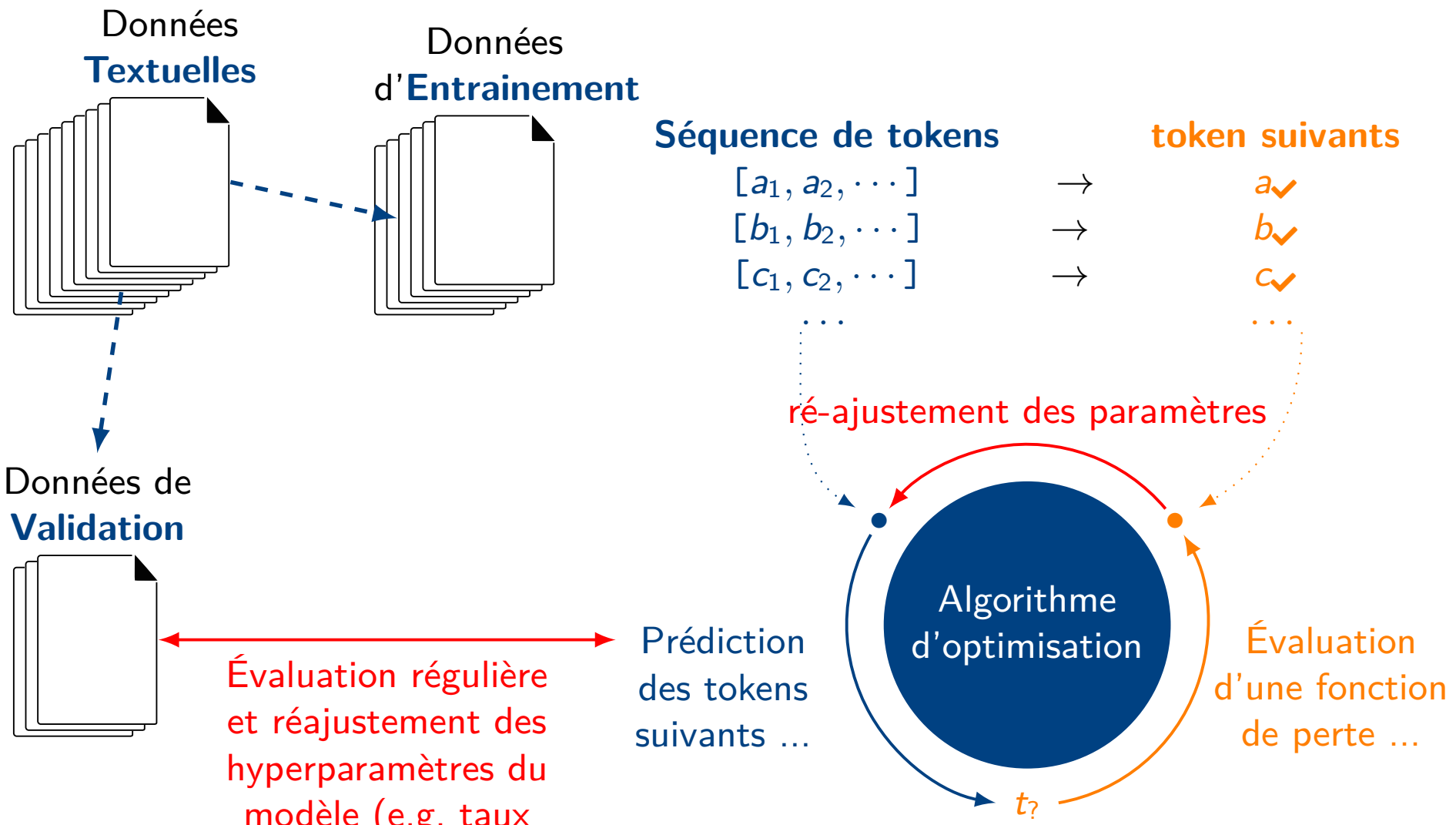


Table des matières

Différentes architectures de « *Transformers* »

Transformer original [1] (Google) : Approche : *blocs d'encodeurs et décodeurs empilés, couche d'attention multi-têtes, réseaux entièrement connectés.*

BERT [2] (Google) : *Bidirectional Encoder Representations from Transformers* Approche : *représentations bidirectionnelles des mots en utilisant le masking de tokens.*

GPT (OpenAI) : *Generative Pre-trained Transformer*
Approche : *modèle autoregressif de génération de texte.*

XLNet [3] (Google) : Approche : *permutationnelle permetant une prédiction de token à partir de n'importe quel autre de la séquence.*

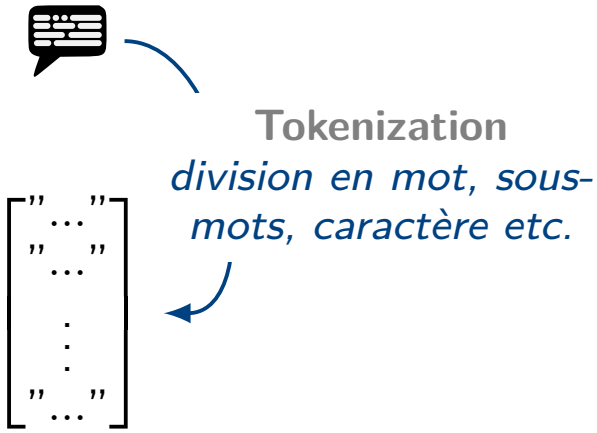
T5 et T5X [4] (Google) : *Text-to-Text Transfer Transformer*
Approche : *Formulation des tâches de TAL sous forme de tâches de traduction de texte (flexibilité, généricité).*

RoBERTa [5] (Facebook) : *Robustly optimized BERT approach*
Approche : *amélioration de BERT à l'aide d'un entraînement plus lourd*

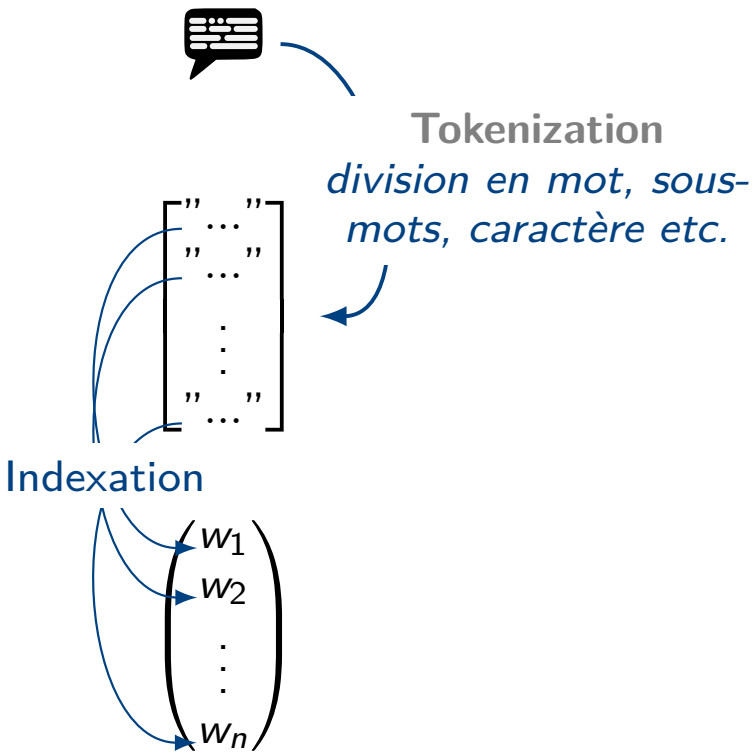
1. Vaswani et al., "Attention is All you Need".
2. Devlin et al., BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding.
3. Yang et al., XLNet : Generalized Autoregressive Pretraining for Language Understanding.
4. Roberts et al., Scaling Up Models and Data with `t5x` and `seqio`.
5. Liu et al., RoBERTa : A Robustly Optimized BERT Pretraining Approach.

Architecture des LLM

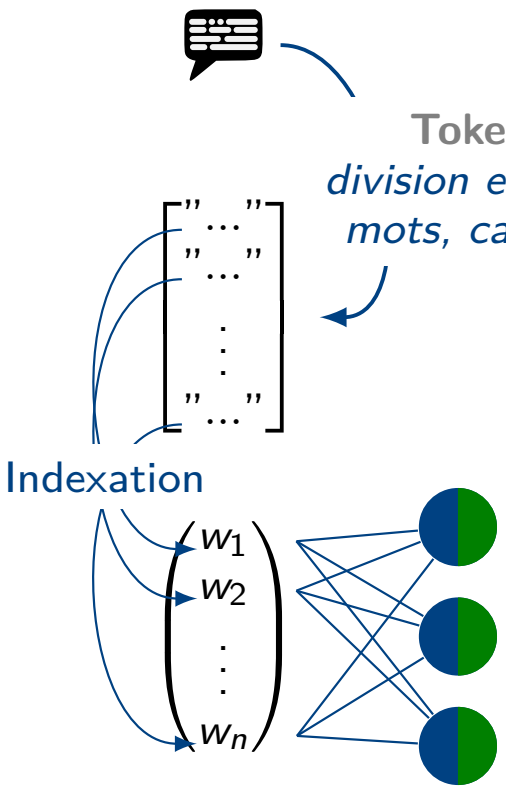
Architecture des LLM



Architecture des LLM



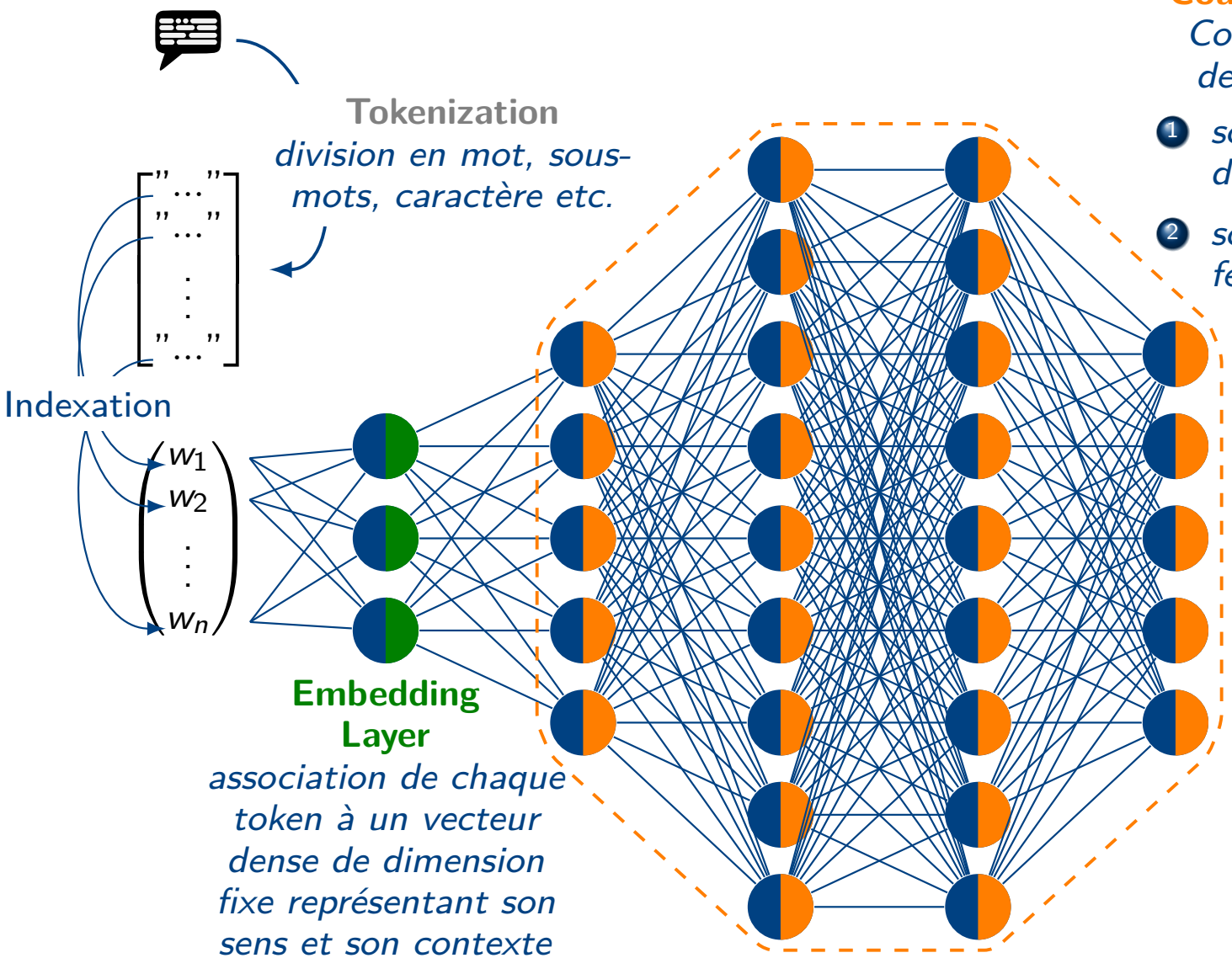
Architecture des LLM



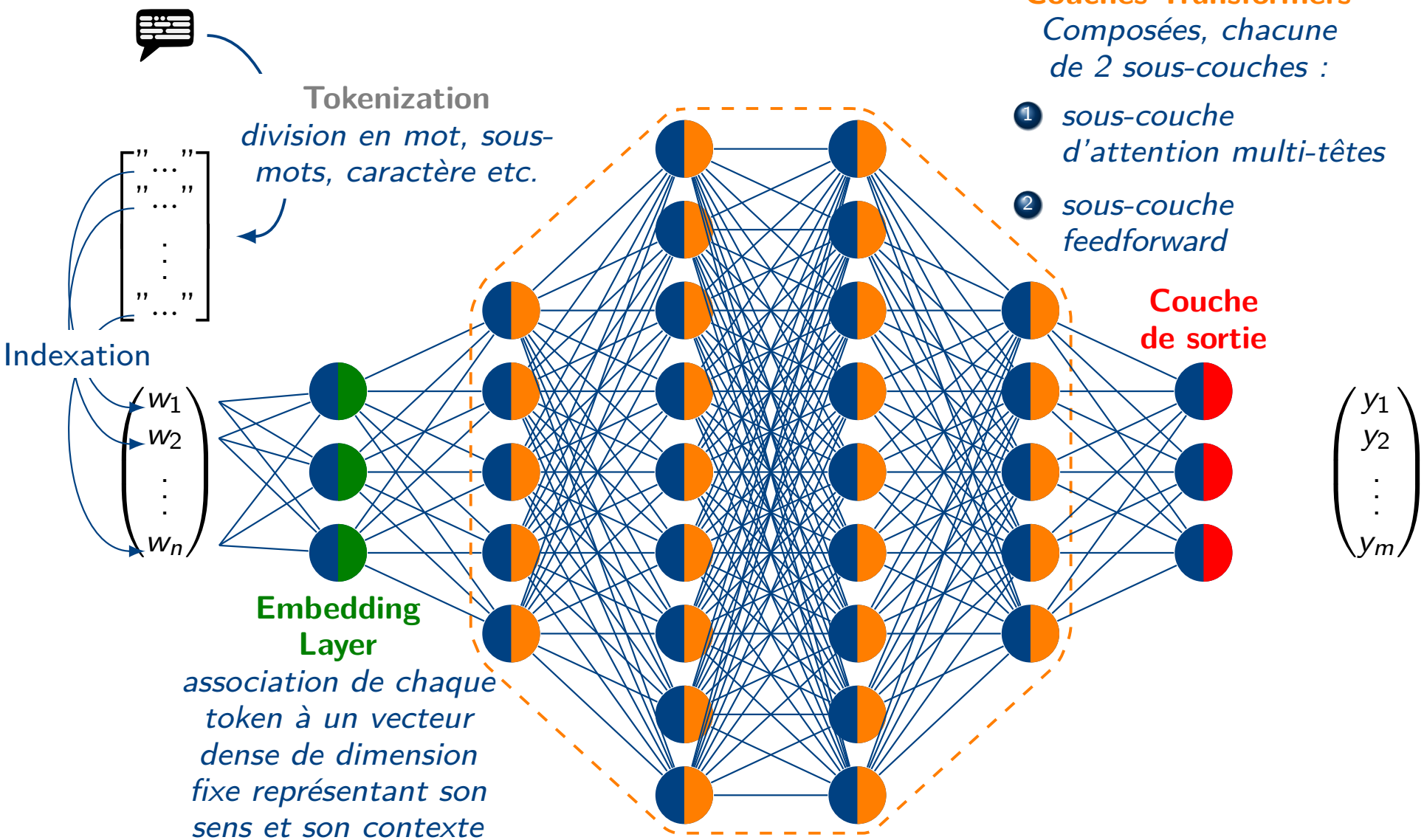
Embedding Layer

association de chaque token à un vecteur dense de dimension fixe représentant son sens et son contexte

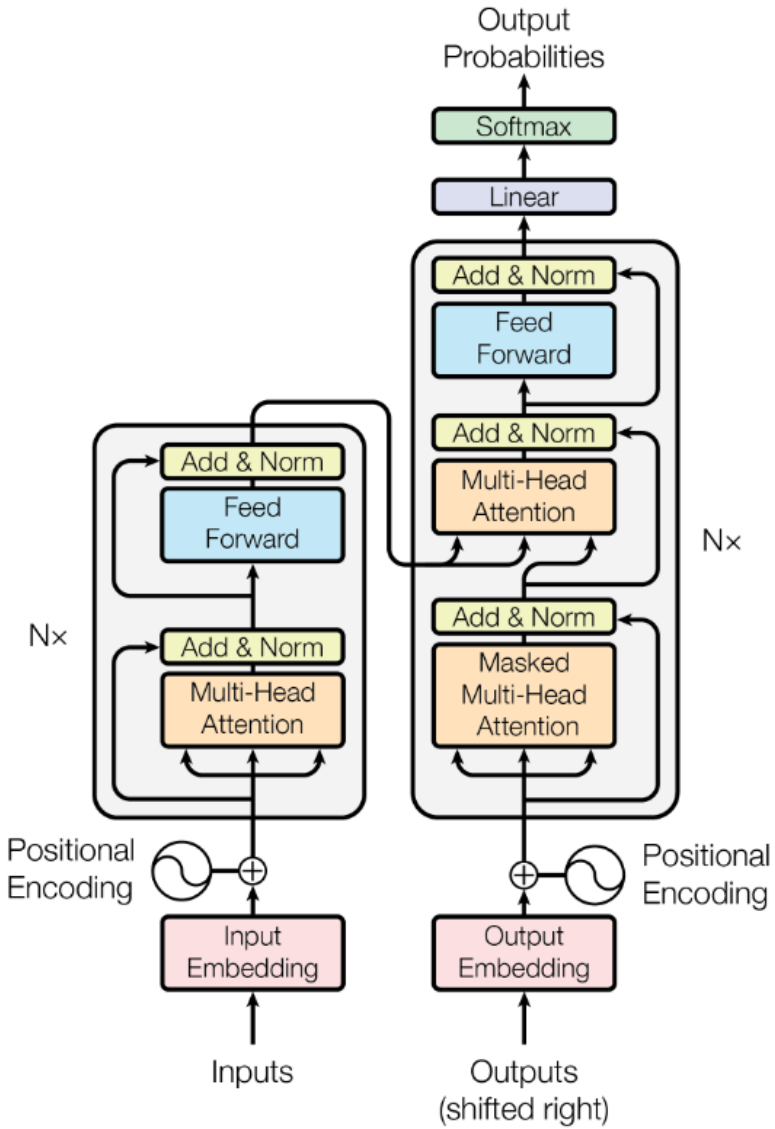
Architecture des LLM



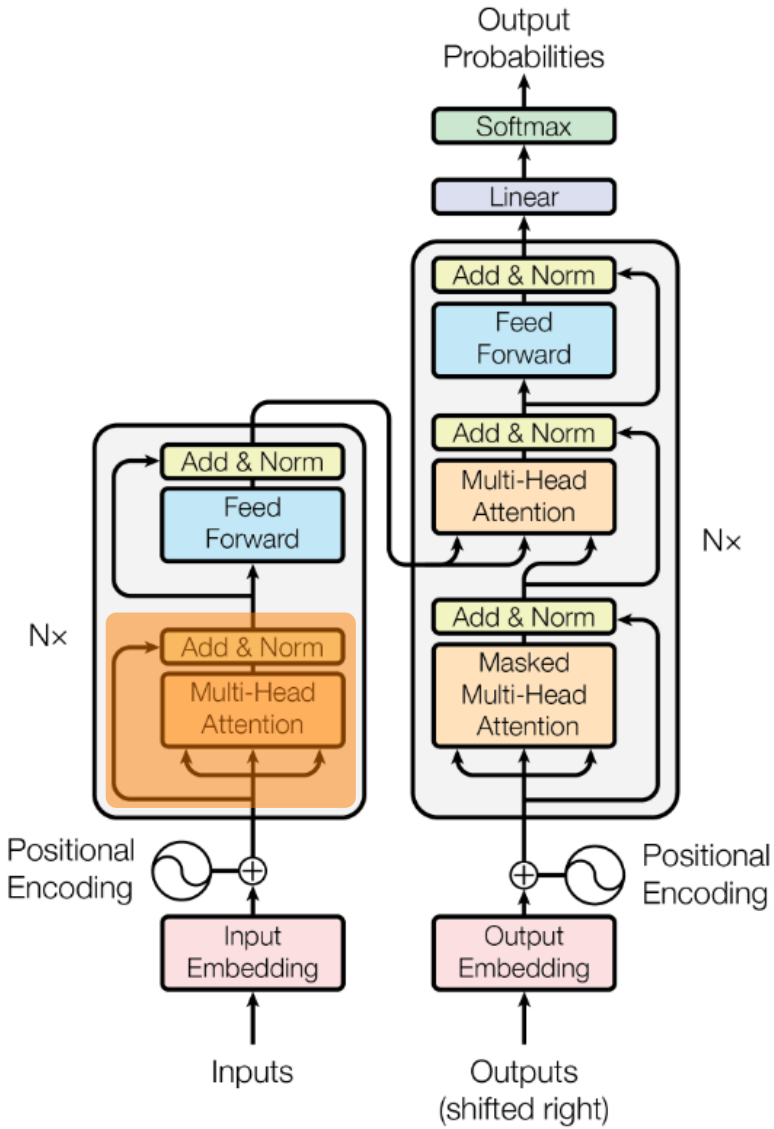
Architecture des LLM



VASWANI et al., "Attention is All you Need"



VASWANI et al., "Attention is All you Need"



Sous-couche d'attention Multitête

Capture les dépendances à longue distance et les relations complexes entre les tokens de la séquence en générant des représentations pondérées de chaque token en fonction de son importance et de sa pertinence dans la séquence globale

VASWANI et al., "Attention is All you Need"

Sous-couche feedforward

Transforme les représentations intermédiaires produites par la sous-couche d'attention multi-têtes en des représentations plus riches et complexes, en appliquant des transformations non linéaires à l'aide de réseaux de neurones entièrement connectés (feedforward)

Sous-couche d'attention Multitête

Capture les dépendances à longue distance et les relations complexes entre les tokens de la séquence en générant des représentations pondérées de chaque token en fonction de son importance et de sa pertinence dans la séquence globale

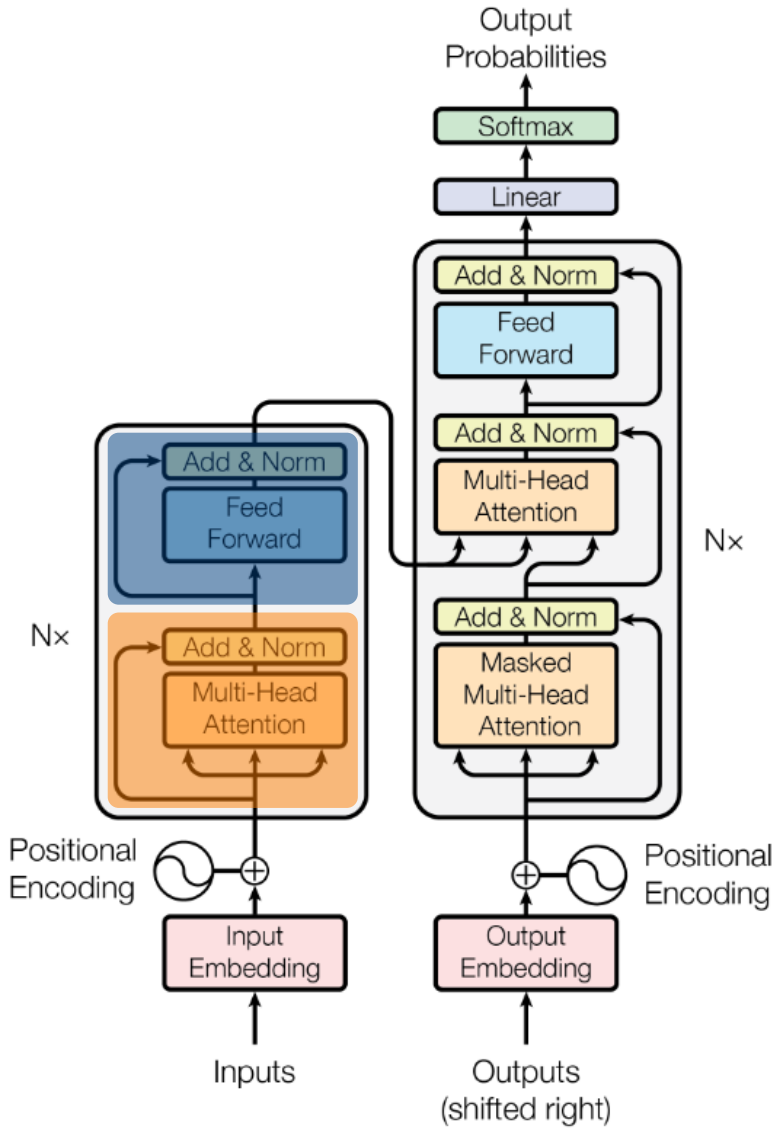
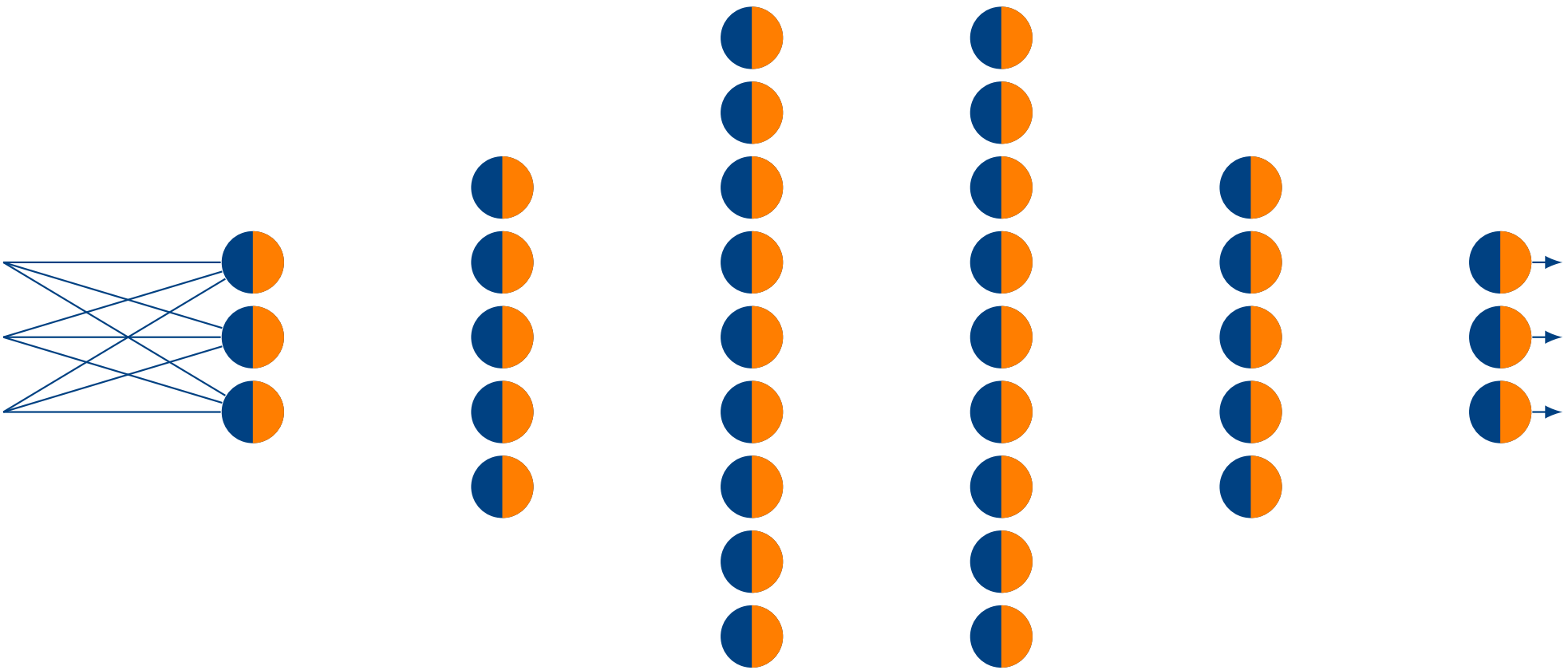


Table des matières

Propriété (Complexité d'un LLM)

Le **nombre de paramètres** d'un LLM correspond au nombre de liaisons entre les neurones. Il constitue un indicateur de sa **complexité**.

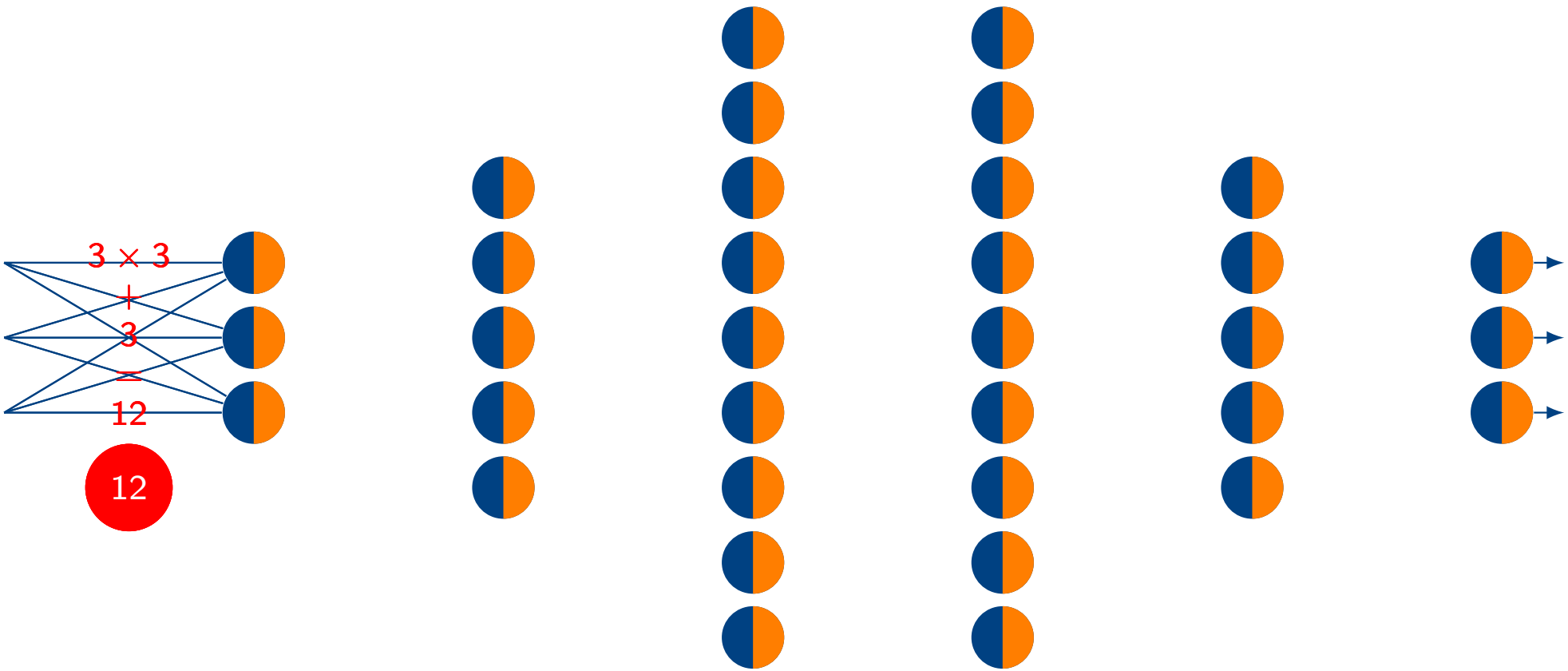
Exemple : dans le cas d'un réseaux complétement connecté ...



Propriété (Complexité d'un LLM)

Le **nombre de paramètres** d'un LLM correspond au nombre de liaisons entre les neurones. Il constitue un indicateur de sa **complexité**.

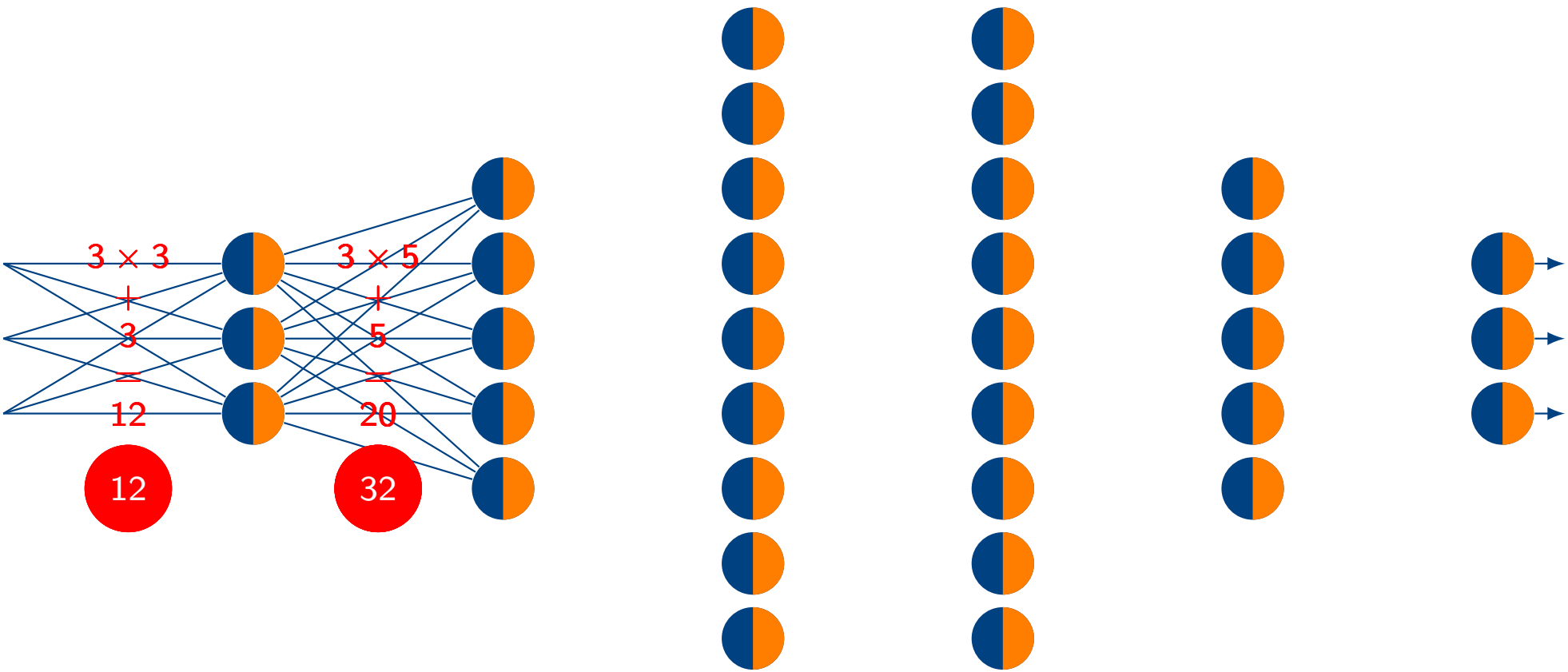
Exemple : dans le cas d'un réseaux complétement connecté ...



Propriété (Complexité d'un LLM)

Le **nombre de paramètres** d'un LLM correspond au nombre de liaisons entre les neurones. Il constitue un indicateur de sa **complexité**.

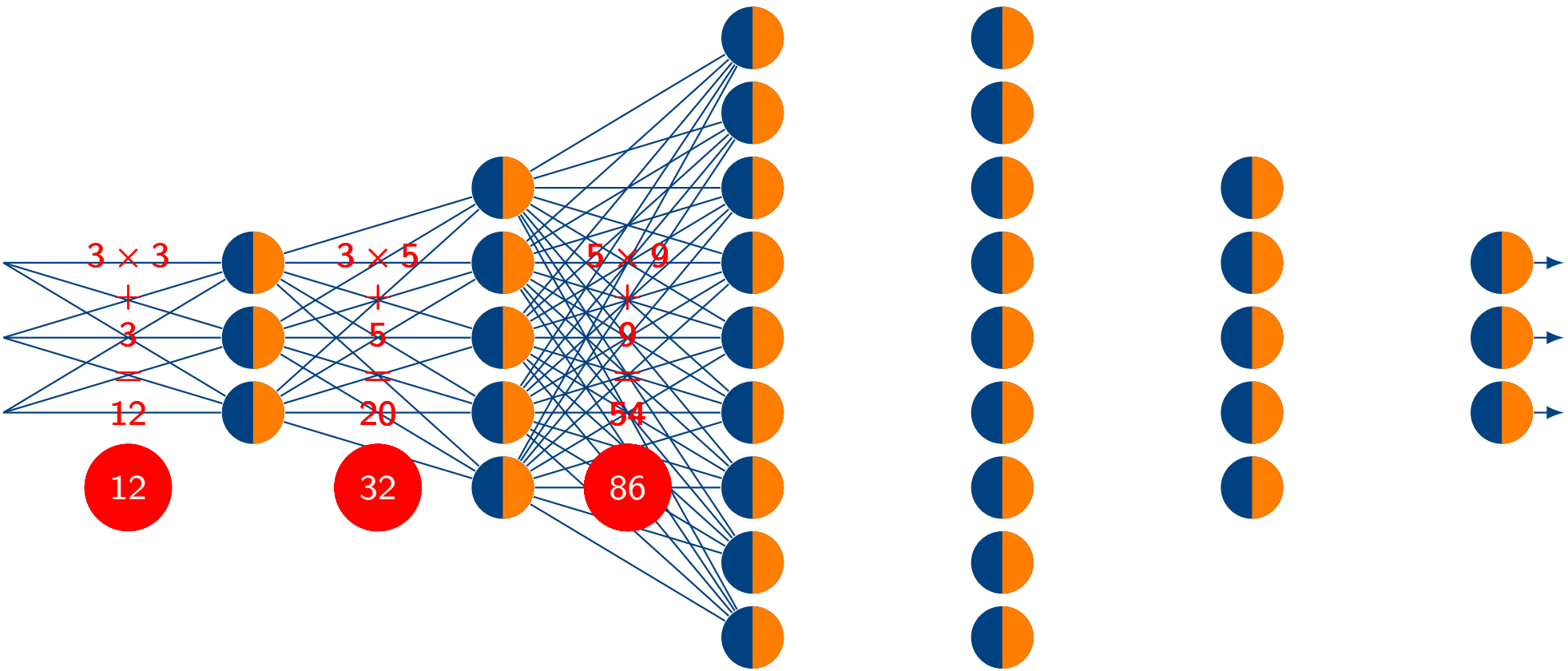
Exemple : dans le cas d'un réseaux complétement connecté ...



Propriété (Complexité d'un LLM)

Le **nombre de paramètres** d'un LLM correspond au nombre de liaisons entre les neurones. Il constitue un indicateur de sa **complexité**.

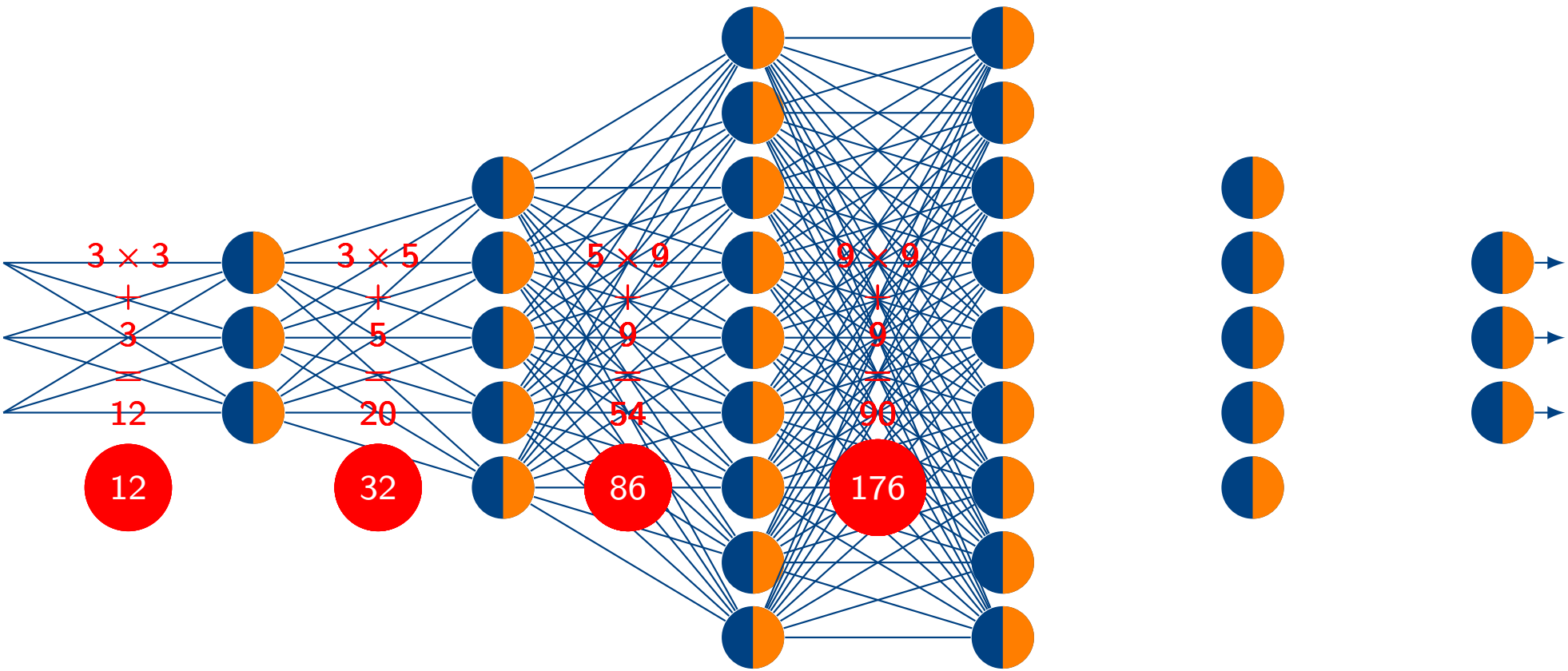
Exemple : dans le cas d'un réseaux complétement connecté ...



Propriété (Complexité d'un LLM)

Le **nombre de paramètres** d'un LLM correspond au nombre de liaisons entre les neurones. Il constitue un indicateur de sa **complexité**.

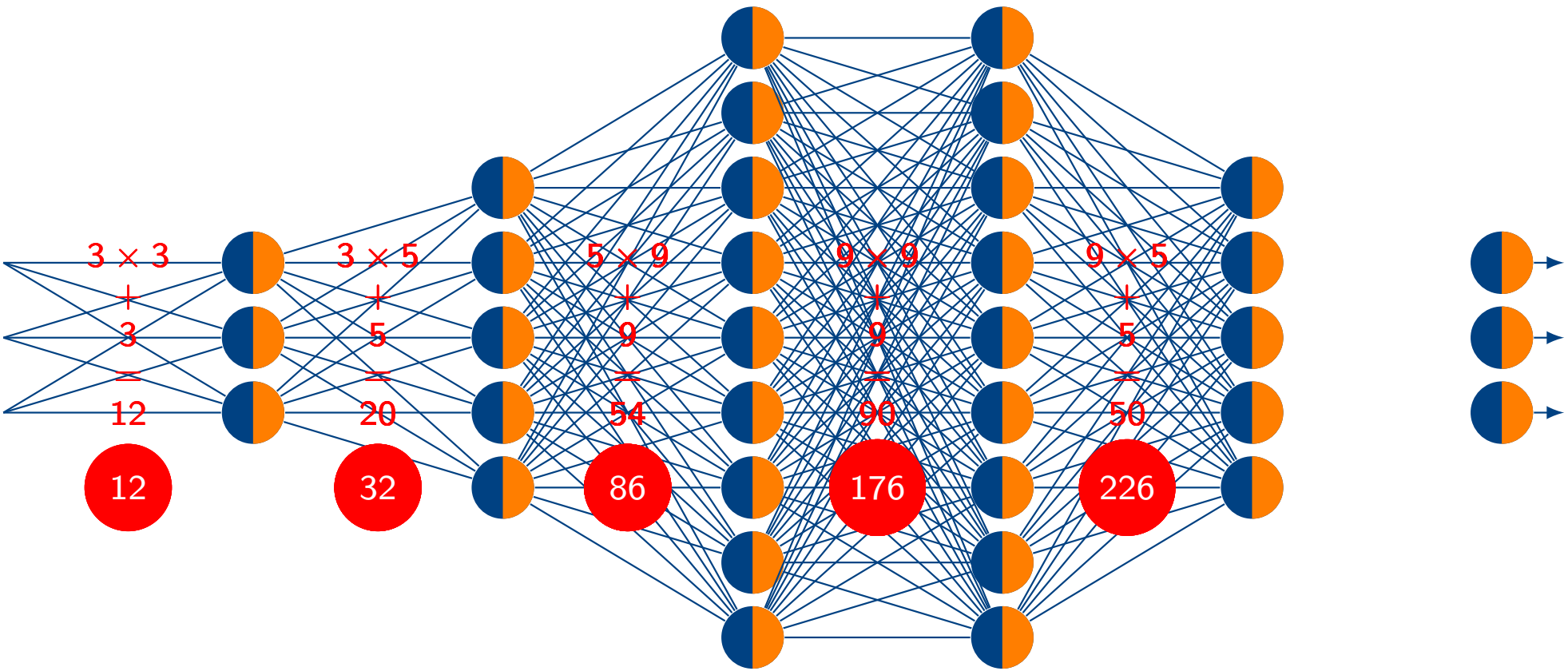
Exemple : dans le cas d'un réseaux complétement connecté ...



Propriété (Complexité d'un LLM)

Le **nombre de paramètres** d'un LLM correspond au nombre de liaisons entre les neurones. Il constitue un indicateur de sa **complexité**.

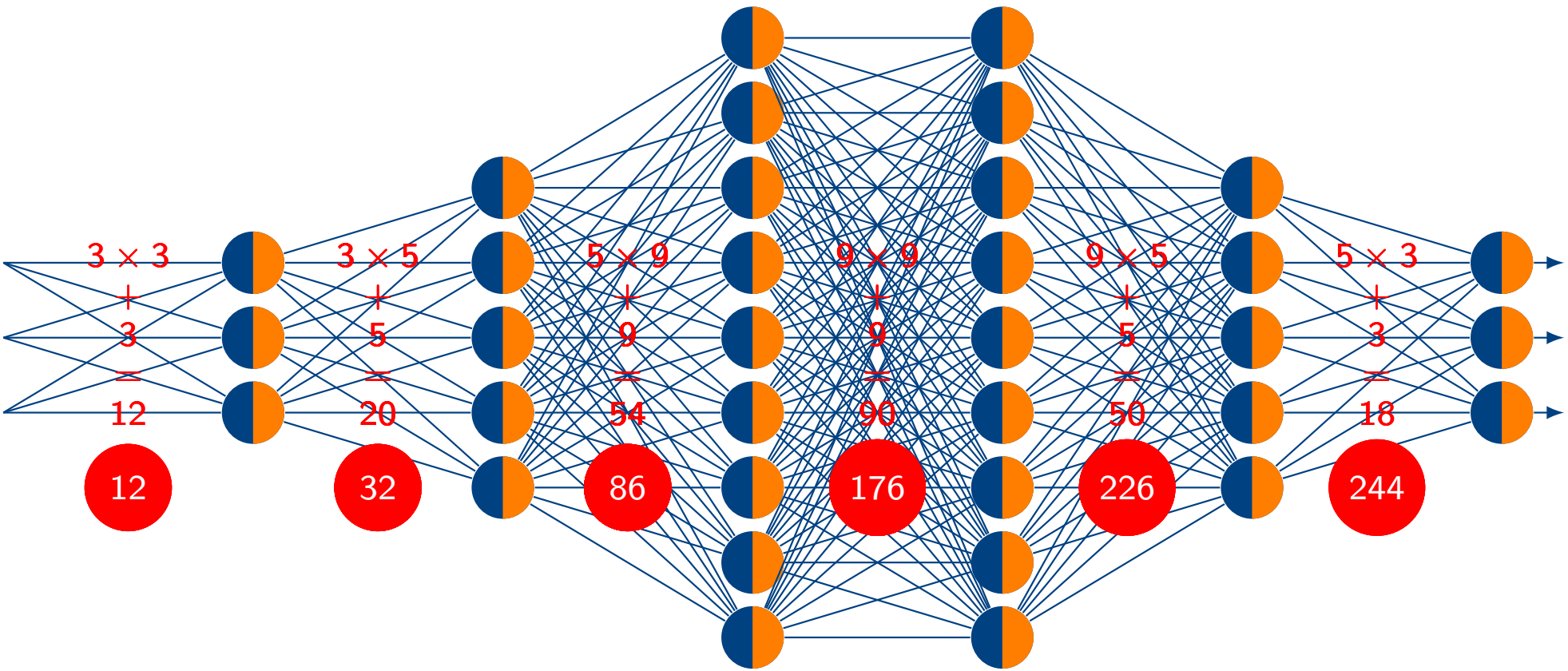
Exemple : dans le cas d'un réseaux complétement connecté ...



Propriété (Complexité d'un LLM)

Le **nombre de paramètres** d'un LLM correspond au nombre de liaisons entre les neurones. Il constitue un indicateur de sa **complexité**.

Exemple : dans le cas d'un réseaux complétement connecté ...



Quelques LLM et leurs nombre de paramètre

Créateur	Date	Nom	Nombre de paramètres	
Google	?	Gemini	?	
Open AI	2023	GPT-4	?	
Google	2022	PaLM	540 000 000 000	
AI21 Labs	2021	Jurassic-1	178 000 000 000	
Hugging Face	2022	BLOOM	176 000 000 000	
Open AI	2020	GPT-3	175 000 000 000	
LLaMa 2	2023	Meta (Facebook)	70 000 000 000	
Vigogne	2023	(LLaMa 2)	13 000 000 000	
LightOn	2022	Lyra-fr	10 000 000 000	
Open AI	2023	GPT4All	7 000 000 000	
DrBERT	2023	Recherche[6]	7 000 000 000	
Mistral AI	2023	Mistral 7B	7 000 000 000	
Google	2018	BERT	340 000 000	

6. Labrak et al., "DrBERT : A Robust Pre-trained Model in French for Biomedical and Clinical domains".

Benchmark des LLM

MMLU [7] : *Measuring Massive Multitask language Understanding Compréhension*
linguistique Générale dans 57 tâches incluant « Professional Medicine », « College Medicine », ou encore « Medical Genetics » et « Virology »

DROP [8] : *Discrete Reasoning Over Paragraphs*
Raisonnement et compréhension linguistique.

BIG-bench (Hard) [9] [10] : *Beyond the Imitation Game benchmark*
Diverses tâches difficiles nécessitant un raisonnement en plusieurs étapes

HellaSwag [11] : *Harder Endings, Longer contexts, and Low-shot Activities for Situations With Adversarial Generations*
Raisonnement de bon sens pour des tâches du quotidien

7. Hendrycks et al., "Measuring Massive Multitask Language Understanding".
8. Dua et al., "DROP : A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs".
9. Suzgun et al., Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them.
10. Srivastava et al., Beyond the Imitation Game : Quantifying and extrapolating the capabilities of language models.
11. Zellers et al., "HellaSwag : Can a Machine Really Finish Your Sentence ?"

Table des matières

InExtenso : Évaluation **In**trinsèque et **Ex**trinsèque des biais dans les LLM

Problématique centrale : Les LLM présentent des problèmes éthiques en terme d'**impact environnementaux** et de **reflexion** et **amplification des biais stéréotypés**.

Objectifs de recherche :

- Apporter une meilleure compréhension des sources et mécanismes en jeux et de leurs interconnexions.
- Aborder la problématique globalement sur toute la chaîne de TALN.
- Proposer un corpus et des méthodes pour détecter, analyser et atténuer ces biais dans les modèles de langue du français.

Scope de la recherche : l'aide au **diagnostic des maladies mentales** et l'**extraction d'informations** à partir de dossiers cliniques (inclusion de patient dans les essais cliniques).

Implication de l'équipe : Work Package 3 : ***Évaluer*** *les méthodes développés dans le projet sur des* ***données cliniques réels***.

1. **Fournir des données** :
   1. Corpus de 200 documents annotés manuellement (cancer du sein et avec un biais de genre potentiel + maladies inflammatoires chroniques de l'intestin) ;
   2. Données relatives au diagnostic de maladie mental.
2. **Évaluer les outils développés** dans le cadre du projet sur ces données.