



IG1

La Quantification ou comment compresser les LLMs pour un usage plus responsable



Jean-Philippe Fourès



27 Juin 2025

Disclaimer



- Pas Data Scientist
- Pas Machine Learning Engineer
- Pas éditeur de modèle
- Retour d'expérience de l'IA sur nos usages infra

Problèmes des LLMs

l'IA et notamment les LLMs sont **gourmands**

- Modèles les plus **performants** : modèles les plus **gros** ;
- Consomment des **ressources conséquentes** ;
- Hardware très **difficilement accessible** ;
- Hardware qui coûte **cher**



La solution!



Quantification

Le chausse-pied des LLMs

Technique algorithmique qui **réduit la taille** occupée par les **paramètres** d'un modèle LLM.



LLM qu'es aquò ?

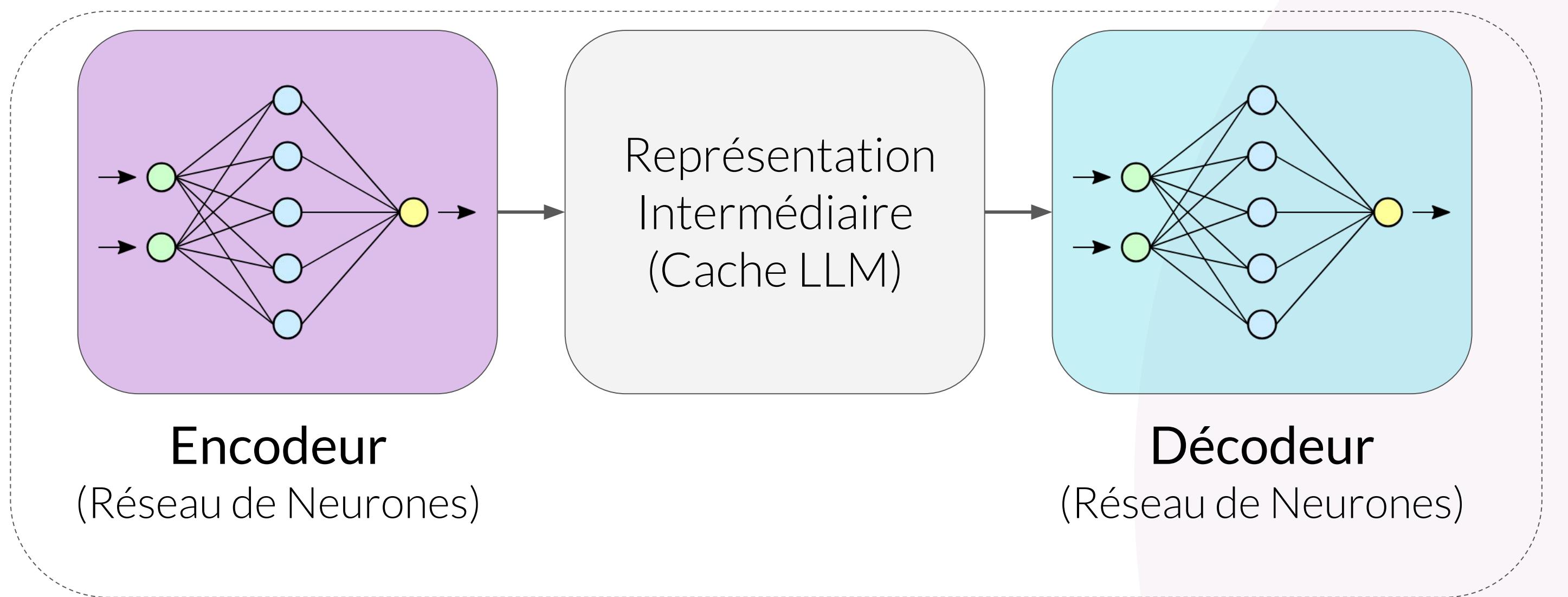
L'architecture courante des LLMs de nos jours :

Transformers!



LLM qu'es aquò ?

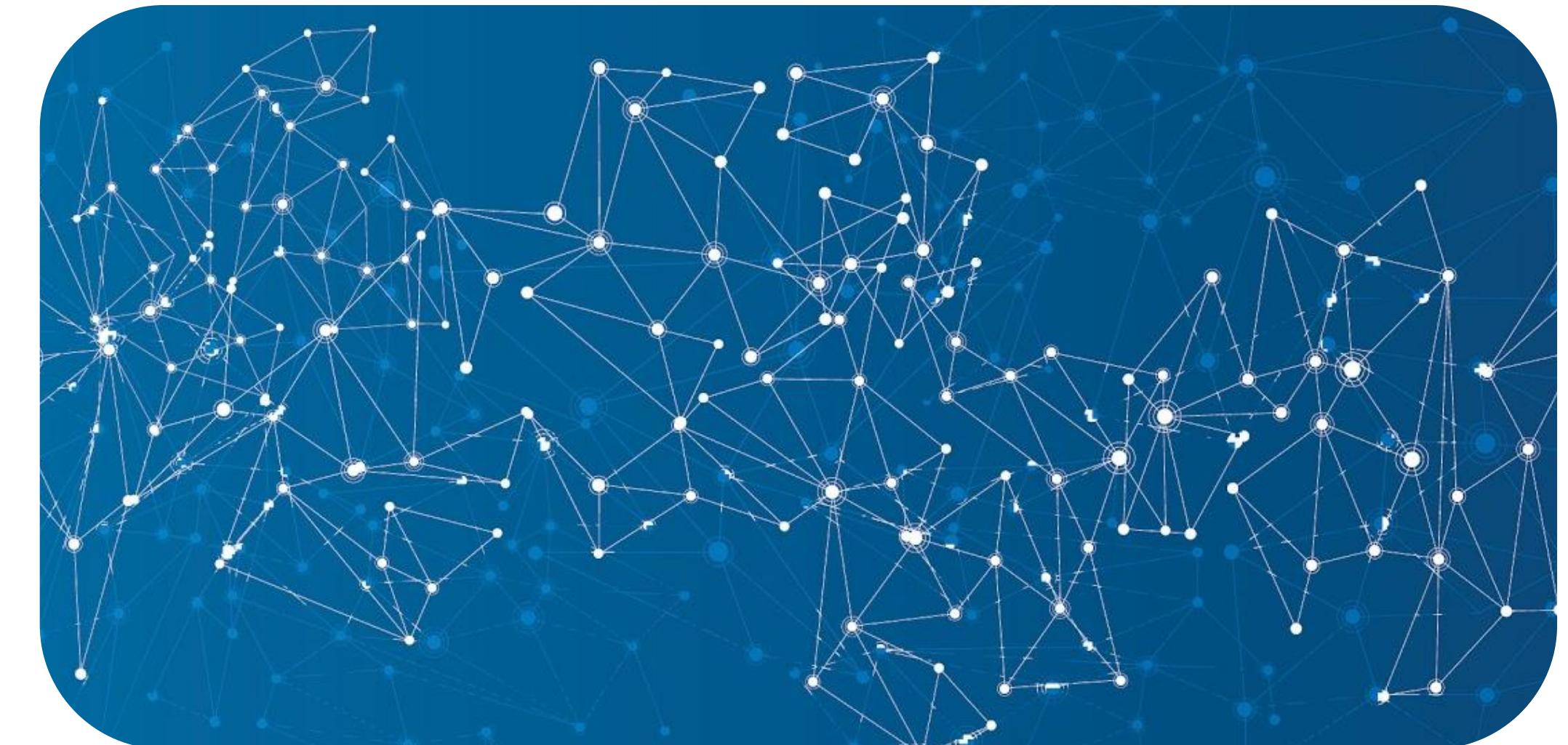
Transformers c'est :



LLM qu'es aquò ?

Chaque neurone est composé de 2 paramètres :

- Poids (**W** pour Weight)
- Fonction d'Activation



Dans un LLM, il y a des millions voire des **Milliards** de paramètres!

Notre Mission

Compressions correctement notre Transformer pour



Notre Mission

qu'il ne devienne pas

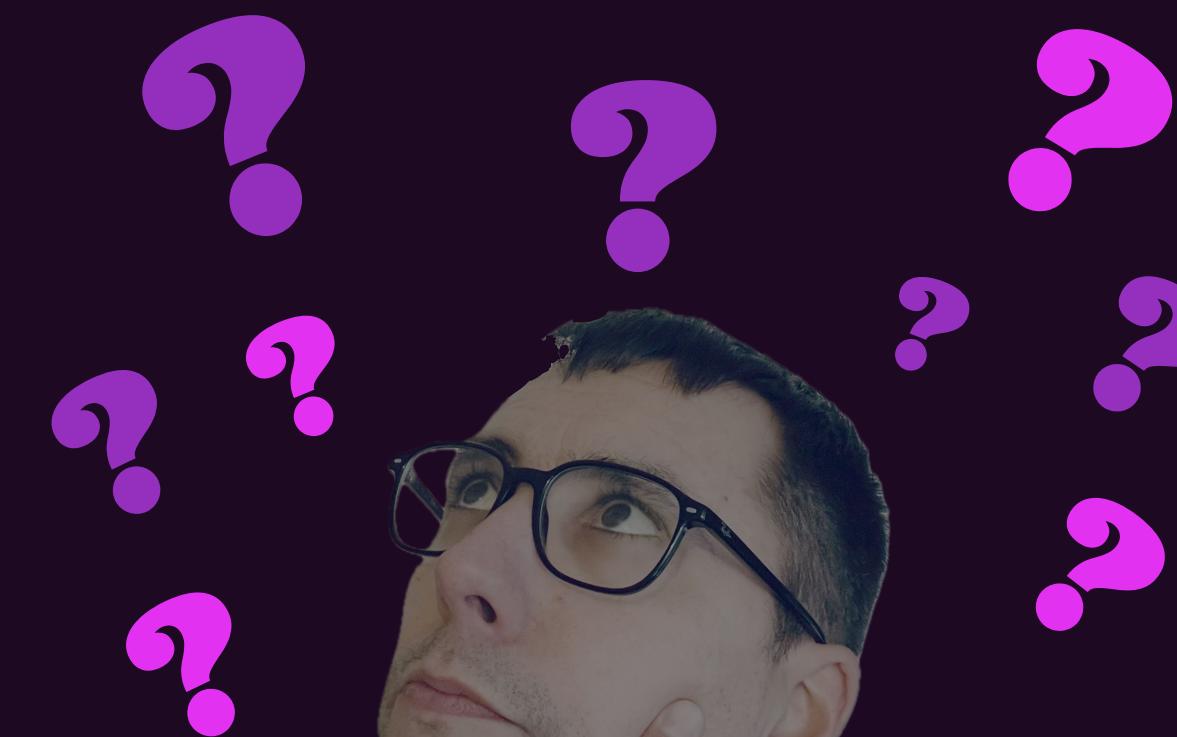


Notre Mission

mais



Principes de Quantification



Comment la quantification réduit
la taille d'un LLM sans (trop)
dégrader sa précision ?

Principes de Quantification

La **compression** génère un fichier **reconstructible**.

La **quantification conserve** le **nombre** de **paramètres**

Mais produit une transformation **irréversible**

en **modifiant le type** des paramètres.

Principes de Quantification

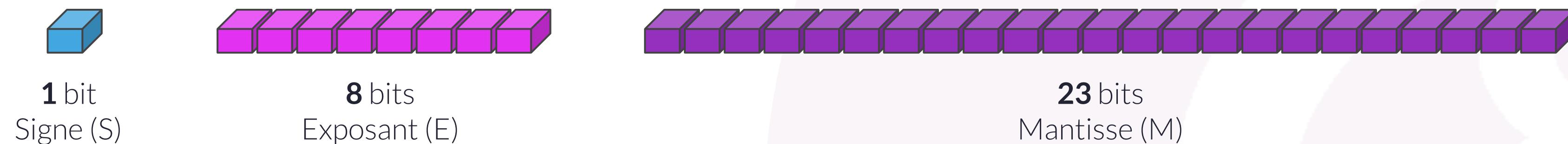
L'**impact** de la quantification sur la **précision** dépend:

- du nouveau **type** choisi pour les paramètres
- de la **technique** de quantification
- du nombre de **paramètres** du modèle.

Types des Paramètres LLM

Virgule flottantes

Le format **FP32** (Floating Point 32 bits) est le plus utilisé pour stocker les paramètres lors de l'**entraînement** des modèles



Types des Paramètres LLM

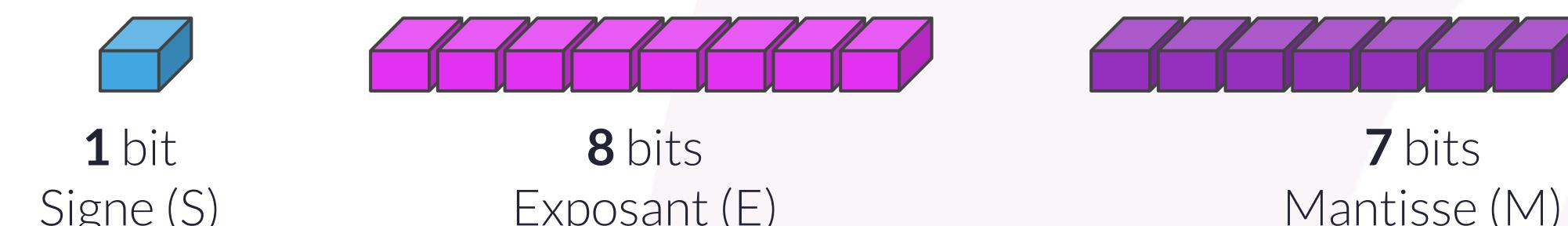
Virgule flottantes

Pour l'**inférence**, les formats les plus fréquents sont :

- FP16:



- BF16:



BF: Brain Floating Point (format inventé par Google Brain pour le Machine Learning)

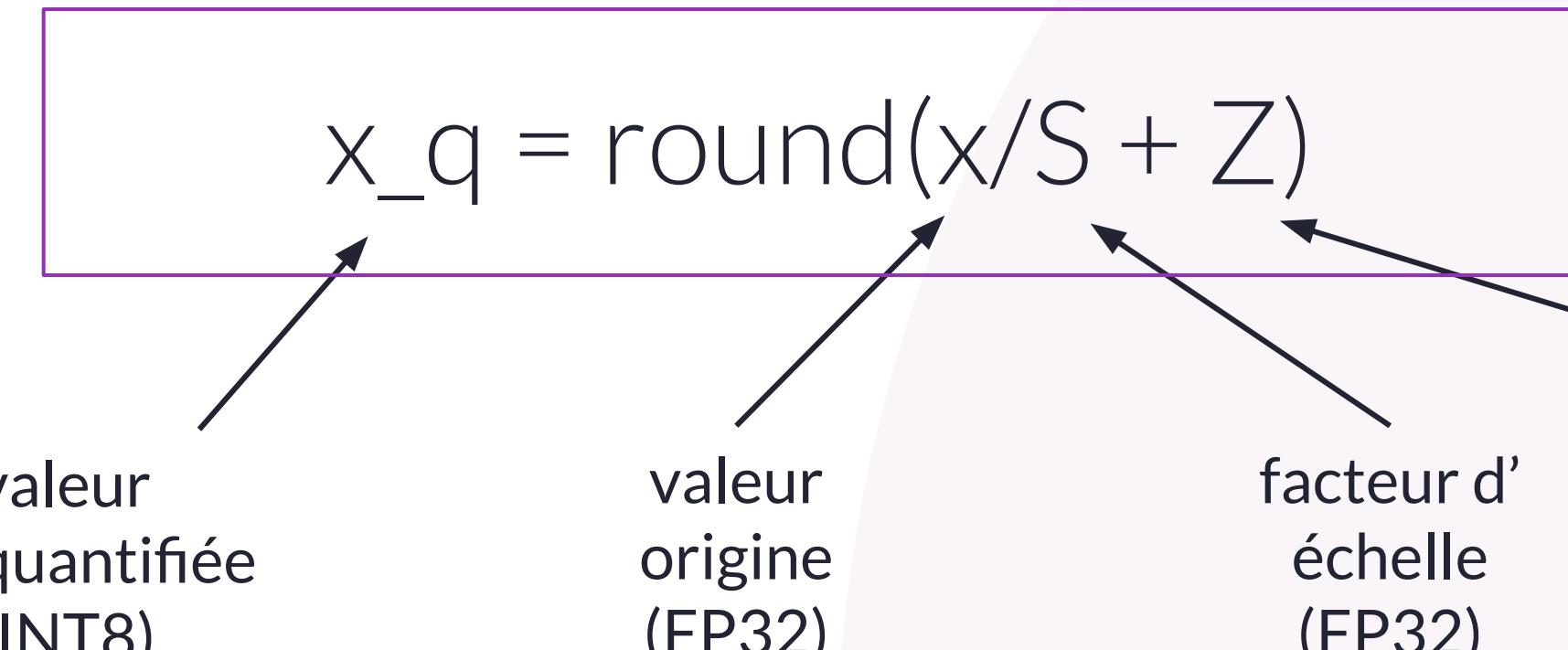
- FP8:



Techniques de Quantification

Quantification Affine (Affine Quantization)

Projection **affine** d'un range $[a, b]$ FP32 sur INT8 (par exemple) :

$$x_q = \text{round}(x/S + Z)$$


The diagram shows the formula $x_q = \text{round}(x/S + Z)$ enclosed in a purple rectangular box. Four arrows point from labels below the box to the corresponding terms in the formula:

- A black arrow points from the label "valeur quantifiée (INT8)" to the term x_q .
- A black arrow points from the label "valeur origine (FP32)" to the term x .
- A black arrow points from the label "facteur d'échelle (FP32)" to the term S .
- A black arrow points from the label "Point 0 du modèle" to the term Z .

Calcul de S et Z: **Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference** :

<https://arxiv.org/abs/1712.05877>

Techniques de Quantification

Quantification AWQ (Activation aWare Quantization)

- Compense le défaut de précision de la quantification Affine
- 99% des poids passent en INT4 (sauf “salient weights”)
- Fonctions d'Activation restent en BF16 ou FP16.
- Gain de taille important (~4) car une majorité des paramètres passe en INT4

Plus d'info sur AWQ: <https://arxiv.org/pdf/2306.00978>

Techniques de Quantification

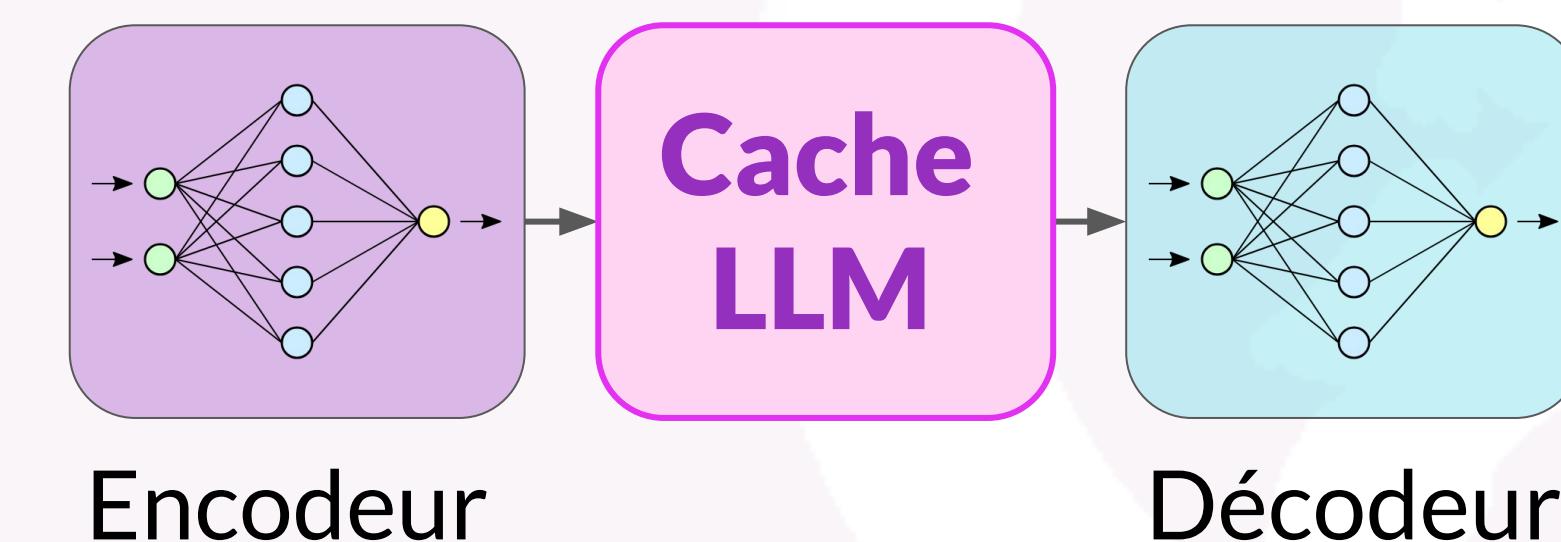
Quantification AWQ - Exemple

Qwen2 72B sur Nvidia **H100** 80GB RAM - **Iguane Solutions**

QWEN 2 72B Instruct BF16	144 GB	3 GPUs
QWEN 2 72B AWQ INT4	41 GB	1 GPU



Pytorch, prompts et contexte chargés dans le **cache LLM** :
20 GB pour **Pytorch**
2.5GB pour **32k tokens**



Techniques de Quantification

Quantification Dynamique FP8

Limites des techniques précédentes:

- Instabilités lors des requêtes à long contexte (> 32k tokens)

Apport de la Quantification dynamique:

- Conversion des Poids **et** Activation en FP8
- Évite les erreurs sur les séquences longues
- Réduction de la méthode Affine avec précision proche d'un modèle FP16.

Techniques de Quantification

Quantification Dynamique (FP8 A8W8) - Exemple

Qwen2.5 72B sur Nvidia **H200** 141GB RAM - **Iguane Solutions**

QWEN 2.5 72B Instruct BF16	144 GB	2 GPUs
QWEN 2.5 72B Instruct FP8	72 GB	1 GPU

55k Token/s

4 000 requêtes/min

Sur 1 seul GPU!

Conclusion

Avantages de la Quantification



Réduction
de la taille avec
conservation du
nombre de paramètres



Calculs plus
simples et plus
rapides



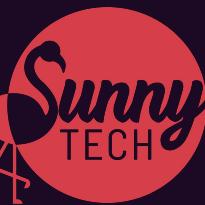
Baisse de la
consommation
énergétique



Utilisation sur
hardware nomade



IG1



MERCI !



Jean-Philippe Foures
VP Product

in @jpfoures **github** @jaypif

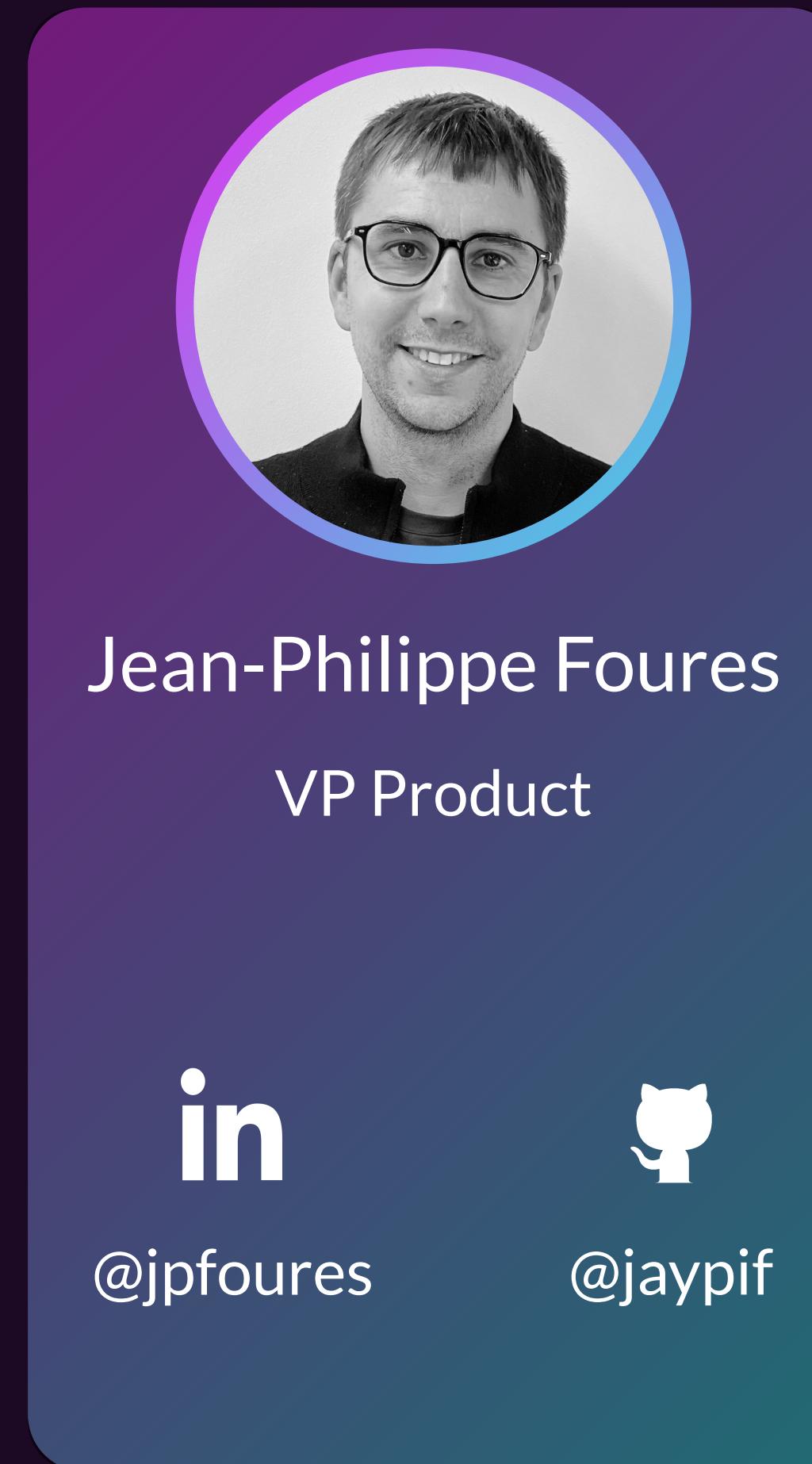


Iguane Solutions

fullstack ai



MERCI !



Vos feedbacks 🙏 :

