AI · MODULE 1 — FONDATIONS TECHNIQUES DES LLM

Qu'est-ce qu'un LLM ?

Chapitre 1 — Introduction aux modèles de langage

Formation développeurs & analystes · INSSE · 45-60 min

OBJECTIFS D'APPRENTISSAGE

Ce que vous saurez faire à la fin

1 Définir

ce qu'est un LLM et expliquer son principe fondamental : prédire le prochain token.

2 Distinguer

pré-entraînement, alignement et inférence — et ce que chacun implique.

3 Expliquer

pourquoi un LLM se trompe, et pourquoi ce n'est pas un bug mais une propriété.

4 Choisir

un modèle de 2026 adapté à un besoin d'entreprise concret.

PLAN DU CHAPITRE

Le parcours du jour

01 L'intuition — une machine à prédire le mot suivant

02 La mécanique — paramètres, probabilités, génération

03 La fabrication — pré-entraînement, alignement, inférence

04 La fenêtre de contexte — la mémoire de travail

05 Les limites — hallucinations, coupure, confidentialité

06 Le paysage 2026 — acteurs, modèles, critères de choix

01

L'intuition

Une machine à prédire le prochain morceau de texte — et tout ce qui en découle.

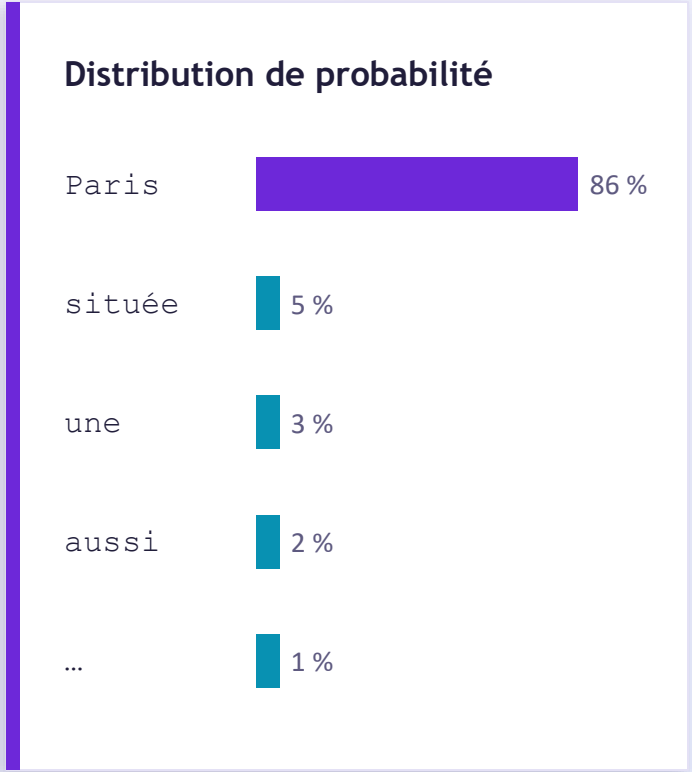
L'INTUITION

Prédire le token suivant, rien d'autre

Un LLM calcule, pour tout contexte donné, **la suite la plus probable**. Comme le clavier prédictif de votre téléphone — mais entraîné sur une fraction énorme du web, des livres et du code.

« La capitale de la France est … »

Le modèle ne récite pas une fiche : il recalcule cette distribution à chaque token généré.



L'INTUITION

Le point contre-intuitif : l'émergence

Ce qu'on lui apprend

- Une seule tâche : prédire le token suivant sur des trillions d'exemples.
- Aucune règle écrite à la main, aucun « si question X alors réponse Y ».

Ce qui émerge

- Grammaire et style dans des dizaines de langues
- Connaissances factuelles générales
- Raisonnement multi-étapes
- Écriture et relecture de code

L'INTUITION

Trois confusions à éliminer d'emblée

Pas une base de données

Il ne stocke ni ne recopie de documents : les connaissances sont diluées dans les paramètres, sans source attachée.

Pas un moteur de recherche

Sans outil externe, il ne consulte rien. Il génère depuis une mémoire figée à l'entraînement.

Pas un programme à règles

Aucun arbre de décision écrit par un humain : une énorme fonction mathématique apprise.

02

La mécanique

Paramètres, distributions de probabilité et génération token par token.

LA MÉCANIQUE

Des nombres, rien que des nombres

Un LLM est un réseau de neurones : des matrices de paramètres ajustés pendant l'entraînement. Le texte entre sous forme de tokens, traverse le réseau, et il en sort une distribution de probabilité sur le vocabulaire.

10^9–10^{11}

paramètres dans les modèles actuels

10^{13}+

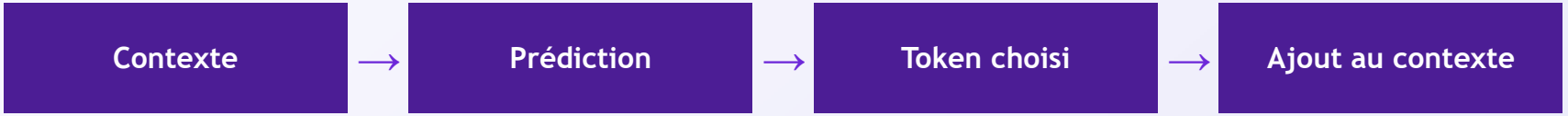
tokens lus au pré-entraînement

~100 k

tokens dans le vocabulaire type

LA MÉCANIQUE

Génération et température



… puis on recommence, des centaines ou milliers de fois — d'où la réponse qui « s'écrit » sous vos yeux.

Température ≈ 0

Choisit presque toujours le token le plus probable. Réponses stables, reproductibles. Pour l'extraction de données, la classification, le factuel.

Température ≈ 1

Échantillonne plus librement dans la distribution. Réponses variées, créatives. Pour le brainstorming, la rédaction, les variantes.

LA MÉCANIQUE

Premier appel LLM en Python

```
import anthropic

client = anthropic.Anthropic()
reponse = client.messages.create(
    model="claude-sonnet-4-6",
    max_tokens=500,
    temperature=0.2,
    messages=[{
        "role": "user",
        "content": "Liste vs tuple en Python ?"
    }]
)
print(reponse.content[0].text)
```

`model`

le modèle choisi — qualité, coût et vitesse en dépendent

`temperature`

0.2 : on privilégie le factuel et le stable

`messages`

tout l'historique est renvoyé à chaque appel : le modèle n'a aucune mémoire entre les appels

03

La fabrication

Du web brut à l'assistant : pré-entraînement, alignement, inférence.

LA FABRICATION

De la matière brute à l'assistant

1 · Pré-entraînement

Lire des trillions de tokens (web, livres, code) et apprendre à prédire la suite. Des dizaines à centaines de millions de dollars de GPU. Résultat : un modèle de base savant mais brut.

2 · Alignement

Fine-tuning d'instructions + retours humains (RLHF) : apprendre à dialoguer, suivre des consignes, refuser le dangereux. Transforme un compléteur de texte en assistant.

3 · Inférence

La phase d'utilisation. Le modèle est figé : vos questions ne le font pas apprendre. Ses connaissances s'arrêtent à la date de coupure.

Figé à l'inférence : deux conséquences

Pas d'apprentissage à l'usage

Corriger le modèle dans une conversation ne le modifie pas : à la conversation suivante, tout est oublié. La « mémoire » d'un produit (préférences, historique) est une couche logicielle ajoutée autour du modèle.

Date de coupure des connaissances

Le modèle ignore tout ce qui est postérieur à son entraînement : versions de frameworks, actualités, vos documents internes. Solutions : recherche web, RAG (chapitre 5), outils (chapitre 6).

04

La fenêtre de contexte

La mémoire de travail du modèle : tout ce qu'il peut « voir » d'un coup.

FENÊTRE DE CONTEXTE

Combien peut-il lire d'un coup ?

Instructions + conversation + documents fournis : tout doit tenir dans la fenêtre. Ce qui dépasse est invisible pour le modèle.

128k

standard d'entrée de gamme (Mistral Large)

200k

Claude Sonnet 4.6 / GPT-5 en standard

1M

tiers premium — ≈ 1 200 pages A4

2M

Gemini : codebase entière en entrée

Pour l'analyste : on peut soumettre contrats, rapports ou bases de code entiers — mais plus de contexte = plus de coût et de latence.

05

Les limites

Hallucinations, coupure, calcul, sensibilité au prompt, confidentialité.

LES LIMITES

Cinq limites structurelles

Hallucination	Le modèle produit le plausible, pas le vrai : s'il ne sait pas, il fabrique avec aplomb. → Chapitre 5.
Date de coupure	Aucune connaissance des événements postérieurs à l'entraînement, sans outil externe.
Calcul non fiable	L'arithmétique est « devinée » comme du texte. Solution : faire appeler des outils. → Chapitre 6.
Sensibilité au prompt	La formulation change la qualité de la réponse. → Chapitre 4 (prompt engineering).
Confidentialité	Ce que vous envoyez part chez le fournisseur : appliquer les règles INSSE sur les données sensibles.

RÈGLE D'OR EN ENTREPRISE

« Le LLM propose, l'humain dispose. »

Toute sortie qui engage l'entreprise — chiffres, droit, code en production — doit être vérifiée par un humain compétent.

06

Le paysage 2026

Quatre champions complémentaires, des modèles ouverts, et des critères de choix.

PAYSAGE

Qui fait quoi en mi-2026

OpenAI — GPT-5.5

Référence sur l'agentique avancée et la polyvalence générale.

Anthropic — Claude Opus 4.7

Tête sur le code (87,6 %) ; accent sûreté. Sonnet 4.6 : 200k–1M de contexte.

Google — Gemini 3.1 Pro

Raisonnement scientifique, rapport perf/prix, contextes jusqu'à 2M de tokens.

Mistral — Large 2.5

Souveraineté européenne, modèles ouverts, hébergement sur site possible.

+ open-weights (Llama, DeepSeek, Mistral) à héberger soi-même · modèles « thinking » : plus lents, meilleurs en maths/logique/code

APPLICATION

En pratique chez INSSE

1 Assistance au code

Revue, tests unitaires, migrations, documentation : gains immédiats côté développeurs.

2 Analyse documentaire

Synthèse de cahiers des charges, comparaison de contrats, extraction de données de rapports.

3 Support & rédaction

Courriels, procédures, traductions FR/EN, premiers jets de livrables.

4 Par où commencer

Cas où l'erreur est peu coûteuse et la vérification facile — puis étendre progressivement.

AVANT DE PARTIR

Quatre pièges courants

1

« Il comprend comme un humain »

Il modélise des régularités statistiques. L'ingénieur raisonne en probabilités, pas en intentions.

2

Demander des faits récents sans outil

Sans recherche ou RAG, la réponse sort d'une mémoire figée — et peut être inventée.

3

Confondre contexte et mémoire

Tout est oublié entre deux conversations ; la fenêtre de contexte n'est qu'une mémoire de travail.

4

Prendre la fluidité pour de l'exactitude

Une réponse bien écrite n'est pas une réponse juste. Vérifier ce qui engage.

À RETENIR

Un LLM est un réseau géant entraîné à prédire le token suivant. De cette tâche simple émergent dialogue, raisonnement et code.

- Fabriqué en deux temps — pré-entraînement puis alignement — utilisé figé en inférence.
- Limité par sa fenêtre de contexte et sa date de coupure ; il peut halluciner.
- Puissant comme assistant, dangereux comme oracle : l'humain vérifie ce qui engage.

La suite du module : tokens & embeddings (ch. 2) · Transformer (ch. 3) · fine-tuning & prompting (ch. 4) · RAG (ch. 5) · agents (ch. 6)