Newsletter N° 8 – Juillet 2025

Les mathématiques derrière les LLM (des IA Génératives)

Les grands modèles de langage (LLM) ne « comprennent » pas comme nous, mais ils apprennent des régularités statistiques à grande échelle. L'objectif de cette Newsletter est de vous montrer, d'une manière simple et "macro", les briques mathématiques essentielles : la tokenisation, l'attention, la fonction softmax, et la manière dont un LLM traite pas à pas une phrase française. L'idée n'est pas de faire de vous un mathématicien, mais de vous donner des repères pour mieux utiliser ces outils, et pour en mesurer les limites.

1) Tokenisation : transformer les mots en « unités » manipulables

Un LLM ne lit pas des mots au sens humain : il transforme le texte en petits morceaux appelés tokens. Selon le tokenizer (BPE, WordPiece, SentencePiece), un token peut être un mot entier (« les »), une sous-unité (comme « magistr » + « ats »), ou un signe de ponctuation. Cette découpe permet de couvrir toutes les langues et tous les mots, y compris ceux qui

n'existaient pas lors de l'entraînement. Chaque token est ensuite converti en un vecteur de nombres (embedding), de dimension typique 768, 1024, 4096…

C'est la « position » du token dans un espace mathématique : des tokens proches sémantiquement ont des vecteurs proches. *(un embedding est un vecteur de nombres qui représente la signification du token dans un espace mathématique).*

Formule — Encodage positionnel (classique Transformer)

Pour indiquer la place d'un token dans la phrase, on ajoute un encodage positionnel sinusoïdal :

$$\mathrm{PE}(pos, 2k) = \sin\left(\frac{pos}{10000^{2k/d}}\right),\ \mathrm{PE}(pos, 2k{+}1) = \cos\left(\frac{pos}{10000^{2k/d}}\right)$$

Légende : PE = Positional Encoding (encodage de position), pos = position du token, k = indice de dimension, d = dimension de l'embedding (vecteur représentant le token).

Un embedding est un vecteur de nombres qui représente la signification du token dans un espace mathématique.

Newsletter N° 8 – Juillet 2025

2) Des vecteurs pour représenter les tokens (embeddings)

Une définition pour commencer : un embedding est un vecteur de nombres qui représente la signification du token dans un espace mathématique).

Pensez à l'embedding comme à une fiche d'identité numérique : il encode, d'une certaine façon, la morphologie, la grammaire et les co-occurrences observées pendant l'entraînement. A cette représentation, on ajoute une information de position pour que le modèle sache si un token vient au début ou à la fin de la phrase. Une des formules classiques de position utilisée dans les Transformers est la suivante (elle alterne sinus et cosinus pour différentes dimensions).

3) L'attention : regarder au bon endroit au bon moment

Le cœur d'un LLM, c'est l'« attention ». L'idée : pour prédire le prochain mot, le modèle calcule, pour chaque token i, à quel point il doit regarder chacun des tokens j du contexte. Il projette chaque token en trois vecteurs : query (pour poser une question), key (pour indexer l'information), value (pour transporter l'information). Le score d'attention entre i et j est obtenu par un produit scalaire normalisé. Ces scores sont ensuite convertis en probabilités via la fonction softmax, puis utilisés pour faire une moyenne pondérée des valeurs. Le résultat est un « contexte » riche pour le token i.

Newsletter N° 8 – Juillet 2025

$$e_{ij} = \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}}$$

Légende : q = query (vecteur de requête), k = key (vecteur clé), d_k = dimension des clés (pour normalisation).

Score d'attention (produit scalaire normalisé).

$$\alpha_{ij} = \text{softmax}_j(e_{ij})$$

Légende : αij = poids d'attention entre le token i et j, softmax = fonction qui transforme les scores en probabilités.

Poids d'attention (probabilités après softmax).

$$\mathbf{c}_i = \sum_j \alpha_{ij} \mathbf{v}_j$$

Légende : ci = vecteur de contexte du token i, vj = value (vecteur valeur) associé au token j.

Vecteur de contexte : moyenne pondérée des valeurs.

4) La fonction softmax : transformer des scores en probabilités

La softmax prend une liste de scores (positifs ou négatifs) et les convertit en une distribution de probabilités qui somme à 1. Elle amplifie les différences : si un score est nettement supérieur aux autres, sa probabilité devient dominante. C'est ce mécanisme qui permet au modèle de « choisir » sur quelles parties du contexte se concentrer.

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Légende : zi = score pour le token i, softmax = normalisation en probabilités.

Newsletter N° 8 – Juillet 2025

5) Entraînement : apprendre à prédire le prochain token

Un LLM est entraîné en essayant de prédire le token suivant, à chaque position, sur des milliards de phrases. On compare la probabilité prédite à la vérité terrain (le véritable prochain token) par une mesure appelée entropie croisée (cross-entropy). Le modèle ajuste ses paramètres pour réduire cette perte, itération après itération. À la fin, il a appris des régularités linguistiques, factuelles et stylistiques qui lui permettent de généraliser. C'est ainsi que sont conçus les phrases : une suite statistique de mots don't le résultat est souvent bluffant.

$$\mathcal{L} = -\sum_{i} y_i \log p_i$$

Légende : yi = valeur attendue (vrai label), pi = probabilité prédite par le modèle.

6) Exemple "pas à pas" sur une phrase simple

Prenons la phrase : " les conseillers prud'homaux sont t-ils de vrais magistrats et jugent-ils en droit ". Nous allons voir comment un LLM typique

la traite, de la découpe en tokens jusqu'au calcul des poids d'attention, pour aboutir à une réponse structurée.

Etape A — Normalisation et tokenisation

La chaîne est normalisée (minuscules, gestion des apostrophes, tirets insécables). Un tokenizer à sous-mots de type BPE pourrait produire une séquence approximative comme : [« les », « conseillers », « prud », « ' », « hom », « aux », « sont », « t », « - », « ils », « de », « vrais », « magistrats », « et », « jugent », « - », « ils », « en », « droit »]. Ce n'est qu'un exemple : selon le modèle, « magistrats » pourrait être découpé en « magistra » + « ts », etc. Chaque élément est immédiatement converti en un vecteur (embedding[1]) et reçoit une position.

Etape B — Représentations Q, K, V et scores d'attention

Pour chaque token, le modèle calcule Q (query), K (key) et V (value). Le token « magistrats » va, par exemple, regarder fortement « vrais » et « de », car ces mots influencent la qualification. Le fragment « t-ils » va pointer vers « sont » et « jugent-ils » pour capter la structure interrogative. Les scores d'attention sont convertis en probabilités grâce à la softmax, puis utilisés pour combiner les V : on obtient pour chaque position un contexte pondéré qui capture la dépendance longue distance.

[1] un embedding est un vecteur de nombres qui représente la signification du token dans un espace mathématique

Newsletter N° 8 – Juillet 2025

Etape C — Empilement de couches et sortie

Plusieurs couches d'attention et de transformations non linéaires se succèdent (multi-têtes, normalisations, feed-forward).

A la fin, le modèle calcule une distribution de probabilité sur le prochain token (ou, en mode instruction, sur la prochaine séquence de sortie). Les tokens les plus probables formeront la réponse.

Dans un cadre juridique, un modèle bien instruit cherchera à définir : qui sont les conseillers prud'homaux, leur statut, leur rôle, et la distinction entre magistrats professionnels et juges non professionnels.

Etape D — Pourquoi ce n'est pas « magique »

Rien n'est deviné : le modèle exploite des régularités apprises. Si ses données d'entraînement contiennent des sources fiables (textes officiels, doctrine, jurisprudence), il a plus de chances de produire une réponse correcte. D'où l'importance du "Data Labelling", c'est à dire "létiquetage de données", c'est l'opération qui consiste à **annoter des données brutes** (texte, image, vidéo, audio, etc.) pour les rendre **compréhensibles par une IA**[2].

[2] Toutes les IA ont recours au Data Labelling sinon elles ne pourraient pas fonctionner.

A l'inverse, sans garde-fous, il peut « halluciner ». D'où l'intérêt d'exiger des citations, d'expliciter les incertitudes, et de verifier les informations surtout lorsqu'elles sont sensible.

7) Le moteur des LLM : les mathématiques linéaires et les probabilités

En conclusion à cette Newsletter nous pouvons dire que les disciplines mathématiques à la base des LLM sont 2 :

1) **Les mathématiques linéaires** (l'algèbre linéaire) = le moteur des LLM.

Newsletter N° 8 – Juillet 2025

2) **Les probabilités** = le carburant de la décision.

A – l'algèbre linéaire

Les LLM manipulent des **vecteurs** et des **matrices**.
Exemple : un mot → transformé en vecteur (embedding).
Ensuite :

- multiplications matricielles ($Q \times K^T$, puis softmax × V),
- combinaisons linéaires,
- transformations dans des espaces de grandes dimensions.

Bref, tout le « calcul brut » qui fait tourner le modèle repose sur l'algèbre linéaire.
C'est la **machinerie** qui manipule les nombres.

B – Les probabilités

Mais à la fin, un LLM doit **choisir le prochain token**.
Et là, on ne parle plus de vecteurs, mais de **probabilités**.
Exemple :

- le modèle calcule que pour la suite d'une phrase :

"magistrat" = 0,62

“avocat” = 0,21

“juge” = 0,17

- Il va tirer (souvent au hasard pondéré) le prochain mot selon ces probabilités.

Les probabilités permettent de :

- interpréter les scores en termes de « chance »,
- introduire de la variété (sampling, température, etc.),
- éviter un déterminisme trop rigide.

C – La relation entre les mathématiques linéaires et les probabilités

En conclusion nous pouvons dire que :

- **L’algèbre linéaire calcule les scores** → des nombres sans signification directe.

Newsletter N° 8 – Juillet 2025

- **Les probabilités transforment ces scores** → en une distribution exploitable (grâce à softmax).

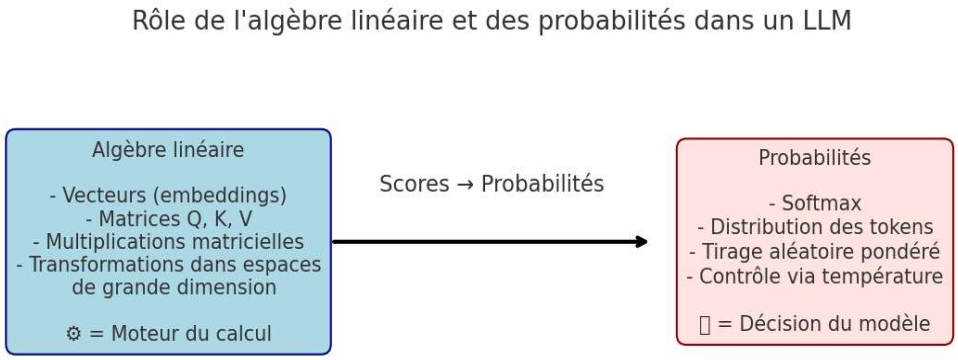
Sans algèbre linéaire, pas de calcul de contexte.
Sans probabilités, pas de choix de sortie.

Métaphoriquement, appliqué à un tribunal nous pouvons dire que :

- **L'algèbre linéaire** = tout le travail d'analyse et de mise en balance (preuves, arguments).

- **Les probabilités** = le délibéré qui tranche : « il y a 70 % de chances que la décision soit celle-ci ou celle là ».

Newsletter N° 8 – Juillet 2025



Que penser de tout cela ?

Vous disposez maintenant d'une cartographie claire : des tokens convertis en vecteurs, une "attention" qui sélectionne les informations pertinentes, une softmax qui transforme des scores en probabilités, et un entraînement par entropie croisée pour s'ajuster.

Retrouvez toutes nos Newsletters sur www.gpappai.com

Newsletter N° 8 – Juillet 2025

Dans la pratique, cette mécanique statistique fonctionne remarquablement bien, mais elle n'est pas infaillible. Des "hallucinations" peuvent intervenir ou une autre source d'erreur : le sur-apprentissage.

Suivant votre demande, nous appliquerons ce cadre pour formuler une réponse à la question posée sur le statut des conseillers prud'homaux et leur qualité de magistrats, en respectant la méthode : définitions, sources, et limites, mais ceci dans une prochaine Newsletter.

Les LLM sont un **enjeux majeur** pour toutes les IA, qu'elles qu'elles soient, nous sommes à la croisée de 2 disciplines : les mathématiques linéaires et les probabilités, en cela les "développeurs" des IA sont essentiellement des mathématiciens. Cela peut paraître abscons, mais ce qui est fascinant c'est précisément de fournir un résultat souvent très pertinent, et plus le temps passera, plus la precision des IA s'affinera.