UN ESSAI INTERACTIF · 1972 → 2026

La mécanique de l'intelligence

L'intelligence artificielle est passée d'une curiosité académique à l'axe autour duquel tournent désormais l'industrie, l'énergie et la géopolitique, et l'essentiel de ce basculement s'est joué en quinze ans à peine. Cet essai tente d'expliquer, honnêtement et sans rien supposer d'acquis, ce qui s'est passé, comment ces systèmes fonctionnent réellement, et de démêler ce que l'on sait vraiment de ce que l'on croit savoir.

Août 2026 · Immersion · thethinkingmachine.dev/fr

01 HISTOIRE · 1972 → 2026

Deux hivers et un coup de tonnerre

L'histoire de l'IA n'est pas une ascension régulière. Tout se joue autour d'une question — qu'est-ce que l'intelligence ? — et de deux réponses rivales : dicter des règles à la machine, ou la laisser apprendre par l'exemple. Pendant quarante ans, c'est la plus intuitive (la première) qui l'emporte. Et c'est la mauvaise.

En 1972, demandez à un chercheur en IA comment construire une machine pensante : la réponse va de soi. L'intelligence, c'est la **logique**. On interroge des experts, on consigne leur savoir en règles — *si le patient a de la fièvre et la nuque raide, envisager une méningite* — et on empile assez de règles pour couvrir toutes les situations au monde.

C'était l'école symbolique, et l'idée n'avait rien de naïf. Elle a d'ailleurs produit des systèmes qui diagnostiquaient des infections aussi bien que des spécialistes — et elle a brassé beaucoup d'argent.

L'idée rivale paraissait, en comparaison, déconcertante : ne pas écrire les règles du tout. Construire un réseau d'unités simples, vaguement inspirées des neurones, lui montrer des exemples, et le laisser **trouver les règles lui-même**. Pendant des décennies, cette approche a désespéré ses partisans — il lui fallait deux choses qui n'existaient pas : des océans de données et une puissance de calcul démesurée. Elle perdait, publiquement, contre la logique écrite à la main.

Deux fois, le domaine a trop promis et a perdu ses financements. En 1973, le rapport Lighthill, commandé par Londres, conclut que les réussites de l'IA ne sont valables que sur des problèmes simples : les crédits britanniques s'évaporent. En 1987, l'effondrement du marché des machines Lisp entraîne celui des systèmes experts — le détail est dans la frise. Ce sont les « hivers de l'IA ». Le second a aussi enterré Thinking Machines Corporation, dont le Connection Machine — un ordinateur massivement parallèle — avait fait le bon pari matériel un quart de siècle trop tôt : le parallélisme, sans les données ni l'argent pour le nourrir.

Ce qui a clos le débat n'est pas une percée philosophique, mais du matériel conçu pour le jeu vidéo et un jeu de données bâti sur une conviction que presque personne ne partageait. En 2012, un réseau entraîné sur deux cartes graphiques de jeu vidéo a pulvérisé la compétition de reconnaissance d'images ImageNet, et l'essentiel des acteurs du domaine changeait de camp en à peine vingt-quatre mois. De ce basculement découle presque tout ce qui a suivi : les chatbots, les agents de code, le marché des puces à mille milliards de dollars, les flots de données et d'argent, jusqu'à cette découverte plus étrange encore – la même recette, en plus grand, devient toujours plus intelligente.

Une mise en garde avant de se lancer dans la chronologie. Ce qui suit est une interprétation, pas un compte rendu neutre. Toute frise historique désigne ses vainqueurs après coup, et celle-ci ne fait pas exception : elle suit les idées qui se sont révélées décisives, ce qui n'a pas nécessairement à voir avec la façon dont les décennies ont été vécues de l'intérieur.

Une grande part de ce qui a fait avancer le domaine s'est jouée ailleurs que dans les idées : dans le matériel, l'argent et l'air du temps. Les GPU sont devenus le moteur de l'IA parce qu'une carte graphique, conçue pour le traitement d'image, se trouvait convenir à l'arithmétique matricielle – personne ne l'avait voulu. Des programmes de recherche entiers ont vécu ou péri sur un rapport gouvernemental. Et chaque vague est arrivée dans un tel emballement qu'en retombant, elle a paru donner tort à l'idée elle-même – alors qu'elle ne donnait tort qu'aux promesses. Lisez ce qui suit comme des repères posés sur un terrain que l'on se dispute encore, non comme un itinéraire tracé d'avance.

UNE CHRONOLOGIE INTERACTIVE

L'ère symbolique · 1972 – 1987 — L'intelligence comme logique : des règles écrites à la main, des symboles, et la conviction que la pensée pouvait se programmer directement.

Hivers & statistiques · 1987 – 2011 — Les financements s'effondrent, les promesses se dégonflent — pendant qu'une idée plus discrète mûrit : laisser les machines trouver les règles dans les données, plutôt que de les leur dicter.

La décennie de l'apprentissage profond · 2012 – 2019 — Les réseaux de neurones, enterrés deux fois, se mettent soudain à fonctionner : les données et les GPU ont enfin atteint le niveau qu'exigeait un algorithme vieux de trente ans.

L'ère du passage à l'échelle · 2020 – 2023 — Une étrange loi empirique : agrandissez la même recette, et elle devient toujours plus intelligente. Un même modèle se met à tout faire, au lieu d'être entraîné pour une tâche.

L'ère des agents · 2024 – aujourd'hui — Les modèles ne se contentent plus de répondre : ils agissent — outils, navigation, code — et poursuivent des objectifs pendant des heures, sans supervision.

1972 Naissance des systèmes experts (MYCIN)

À Stanford, MYCIN diagnostique des infections sanguines à l'aide d'environ 600 règles écrites à la main, du type « si… alors… », et égale les spécialistes humains. La leçon qu'on en tire : l'intelligence, ce sont des règles, et avec assez de règles on y arrivera. La même année, le langage de programmation logique Prolog naît à Marseille.

1973 Le rapport Lighthill

Le gouvernement britannique commande au mathématicien James Lighthill un état des lieux de l'IA. Son verdict – grandes promesses, résultats obtenus sur des problèmes minuscules et qui s'effondrent dès qu'on sort du laboratoire – tue l'essentiel des financements britanniques et inaugure un cycle qui se répétera : l'IA oscille entre surpromesse et retour de bâton.

1980 L'IA commerciale : XCON

Digital Equipment Corporation déploie XCON, un système à base de règles qui configure les commandes d'ordinateurs – une machine de l'époque arrivait en caisses de pièces détachées qui devaient s'assembler exactement, et un seul câble mal choisi la rendait muette. Des dizaines de millions de dollars économisés chaque année, dit-on. Toute une industrie de sociétés de « systèmes experts » et de machines Lisp spécialisées s'ensuit.

1986 La rétropropagation sort de l'ombre

Rumelhart, Hinton et Williams publient un exposé limpide de la rétropropagation – l'algorithme qui permet aux réseaux de neurones multicouches d'apprendre de leurs erreurs. Les mathématiques existaient depuis des années ; l'idée a désormais son manifeste. Presque personne ne soupçonne qu'elle fera un jour tourner le monde.

1987 Le deuxième hiver de l'IA

Le marché des machines Lisp s'effondre ; les systèmes experts se révèlent fragiles et coûteux à maintenir. Les financements s'évaporent. « IA » devient un mot que les chercheurs évitent dans leurs demandes de subventions – pendant une vingtaine d'années.

1989 Un réseau lit l'écriture manuscrite

Le réseau de neurones convolutif de Yann LeCun apprend à lire les chiffres manuscrits aux Bell Labs, et finira par traiter une part significative des chèques bancaires américains. La preuve qu'apprendre par l'exemple peut battre l'écriture de règles, au moins dans un domaine étroit.

1997 Deep Blue bat Kasparov

Deep Blue d'IBM bat le champion du monde d'échecs – par la force brute de la recherche et une évaluation réglée à la main, presque sans apprentissage. Un triomphe, mais celui de l'ancien paradigme. La même année, Hochreiter et Schmidhuber publient le LSTM, un réseau de neurones capable de se souvenir dans le temps.

2006 Le « deep learning » reçoit son nom

Geoffrey Hinton et ses collaborateurs montrent comment entraîner des réseaux à nombreuses couches, et rebaptisent le domaine « deep learning » – l'apprentissage profond. Pendant ce temps, les GPU – conçus pour le jeu vidéo – se révèlent accidentellement parfaits pour l'arithmétique matricielle dont les réseaux ont besoin.

2009 ImageNet : le jeu de données comme acte de foi

L'équipe de Fei-Fei Li publie ImageNet : 3,2 millions d'images étiquetées – 14 millions à terme. Le pari – moqué à l'époque – est que ce qui manque n'est pas un algorithme plus malin, mais davantage de données. Une compétition annuelle y est adossée.

2012 Le tournant AlexNet

Un réseau profond entraîné sur deux GPU grand public par Alex Krizhevsky, Ilya Sutskever et Geoffrey Hinton écrase la compétition ImageNet – 15,3 % d'erreur contre 26,2 % pour le deuxième. Dans un domaine où le progrès se mesurait en fractions de pourcent, c'est un coup de tonnerre. En deux ans, toutes les équipes sérieuses de vision par ordinateur passent aux réseaux de neurones.

2014 Des réseaux qui génèrent, et traduisent

Les réseaux antagonistes génératifs (GAN) montrent que les réseaux peuvent créer des images, pas seulement les classer. Les modèles séquence-à-séquence commencent à traduire les langues de bout en bout. Google rachète DeepMind, un laboratoire londonien qui a tout misé sur l'apprentissage.

2016 AlphaGo

AlphaGo de DeepMind bat Lee Sedol au go, un jeu qui compte plus de positions que d'atomes dans l'univers observable – où la force brute est sans espoir et où l'on croyait l'« intuition » indispensable. Le coup 37, qu'aucun humain fort n'aurait joué, devient le symbole de la créativité machine. 280 millions de personnes regardent.

2017 « Attention Is All You Need »

Huit chercheurs de Google proposent une autre façon de traiter le texte : au lieu de le lire mot après mot, dans l'ordre, laisser chaque mot regarder tous les autres à la fois et apprendre lesquels comptent pour lui. Ce mécanisme, c'est l'attention ; l'architecture qui l'emploie, le transformer. L'article lui-même ne parle que de traduction automatique – problème modeste

au regard de ce qui suivra. Sa vraie portée est ailleurs : plus on l'agrandit, meilleure elle devient, sans plafond visible à ce jour. Quasiment tous les systèmes d'IA dont vous avez entendu parler depuis sont des transformers.

2018 Le pré-entraînement change l'économie du domaine

BERT (Google) et GPT-1 (OpenAI) montrent qu'on peut entraîner une seule fois un grand modèle sur du texte brut, puis l'adapter à moindre coût à de nombreuses tâches. La modélisation du langage – prédire le mot suivant – devient discrètement la tâche la plus importante de l'IA.

2020 GPT-3 et les lois d'échelle

OpenAI entraîne sur une large tranche d'internet un modèle de 175 milliards de paramètres – les nombres internes que l'entraînement ajuste, expliqués en détail dans la section suivante. GPT-3 sait écrire, traduire, répondre à des questions : rien de tout cela ne lui a été explicitement appris, ces capacités sont simplement apparues avec l'échelle. L'article des « lois d'échelle » de Kaplan et al. montre que la performance s'améliore comme une fonction lisse et prévisible du calcul, des données et de la taille. L'intelligence, soudain, a une courbe de prix.

2022 ChatGPT

Une interface de chat autour d'un GPT-3.5 affiné, lancée en novembre comme « aperçu de recherche ». 100 millions d'utilisateurs en deux mois – le produit grand public adopté le plus vite de l'histoire à ce moment-là. L'IA cesse d'être un sujet de recherche et devient un fait public.

2023 GPT-4, Claude, et la rébellion des poids ouverts

GPT-4 réussit des examens professionnels et raisonne sur images et texte. Anthropic – fondée par d'anciens chercheurs d'OpenAI soucieux de sûreté – publie Claude. Meta libère les poids de Llama – les réglages appris du modèle –, semant un écosystème open source mondial. Les gouvernements se mettent sérieusement à rédiger des règles pour l'IA.

2024 Les modèles apprennent à réfléchir avant de répondre

o1 d'OpenAI et ses successeurs sont entraînés par renforcement – un apprentissage par récompense et sanction, plutôt que par imitation d'exemples – à dérouler un long raisonnement, étape par étape et hors de la vue de l'utilisateur, avant de donner sa réponse. Ils échangent du temps de calcul contre de la justesse. Un deuxième axe apparaît : non plus seulement des modèles plus gros, mais plus de réflexion par question.

2025 Le choc DeepSeek – et l'année des agents

Le laboratoire chinois DeepSeek publie R1, un modèle de raisonnement à poids ouverts entraîné pour une fraction de ce que coûtaient, croyait-on, les modèles de pointe – effaçant brièvement ~600 milliards de dollars de la valeur de Nvidia en une journée. Pendant ce temps, les « agents » arrivent pour de bon : des modèles qui pilotent des ordinateurs, écrivent et exécutent du code, et utilisent des protocoles d'outils standardisés comme MCP. Claude Code et les outils du même genre font de l'IA un collègue dans le terminal.

2026 Les goals : l'autonomie sur des heures et des jours

Les systèmes de pointe acceptent désormais un objectif durable – « migre cette base de code », « trouve et corrige les vulnérabilités », et le poursuivent par de longues boucles de planification, d'action et d'auto-vérification, confrontant leur propre travail à des tests et à des critères de succès plutôt que d'en référer à un humain à chaque étape. Anthropic sort la famille Claude 5 (Fable), OpenAI la série GPT-5.6, Google Gemini 3.x ; les modèles chinois à poids ouverts (DeepSeek V4, Kimi K3, Qwen 3.6) rattrapent une bonne part du peloton de tête. La question de recherche a glissé de « sait-il répondre ? » à « peut-on lui faire confiance pour agir ? »

LA FORME DE L'HISTOIRE

Remarquez la cadence de cette frise : le succès qui définit chaque ère vient de l'abandon de l'hypothèse centrale de la précédente. Pour l'IA symbolique, le savoir devait s'écrire ; pour l'apprentissage profond (*deep learning*), il devait s'apprendre par l'exemple. La course à la taille supposait que tout se jouait dans des entraînements plus gros ; l'arrivée des agents a ajouté un second levier – laisser le modèle *réfléchir et agir plus longtemps* au moment de l'usage. La forme la plus récente déplace la question plus qu'elle n'y répond : donnez au système un objectif vérifiable, il planifiera, agira, contrôlera son propre travail et itérera des heures durant sans supervision. Chaque bascule, vue depuis le paradigme précédent, ressemblerait à de la triche.

Mais voyez ce que change cette autonomie. L'intelligence cesse d'être jugée à la qualité d'une réponse pour l'être à la fiabilité d'une longue chaîne d'actions. Autre problème, autres modes de défaillance : un modèle qui a raison 95 % du temps impressionne en conversation et devient quasi inutile sur quarante étapes dépendantes, où les erreurs s'accumulent.

L'autonomie, elle aussi, coûte cher – un aspect que les démonstrations laissent rarement voir. Chaque étape de délibération est du temps machine que quelqu'un paie. La même consigne exécutée deux fois peut emprunter deux chemins. Et plus un système agit seul avec compétence, plus il devient difficile de le superviser, de l'auditer ou de l'arrêter. Le problème n'a pas disparu : il s'est déplacé, du résultat vers la supervision elle-même.

Apprendre, c'est régler des millions de curseurs

Ce récit courait jusqu'en 2026. Rembobinons : retirez la couche de vocabulaire, et l'apprentissage automatique (*machine learning*) se résume à un seul principe, répété à une échelle toujours plus grande — un principe qu'on peut saisir en entier, sans zone d'ombre. Décortiquons-le ensemble.

Oubliez « l'intelligence artificielle » un instant. Considérez un objet plus humble : une boîte avec une entrée, une sortie, et quelques millions de réglages — imaginez autant de petits curseurs à faire coulisser. Donnez-lui une photo ; elle sort un nombre entre 0 et 1 qui signifie « à quel point est-ce un chat ? ».

Tant que ses réglages sont pris au hasard, ses réponses ne valent rien. Le domaine entier de l'apprentissage automatique est l'étude d'une seule question : **comment déplacer les curseurs pour que les réponses cessent de ne rien valoir ?**

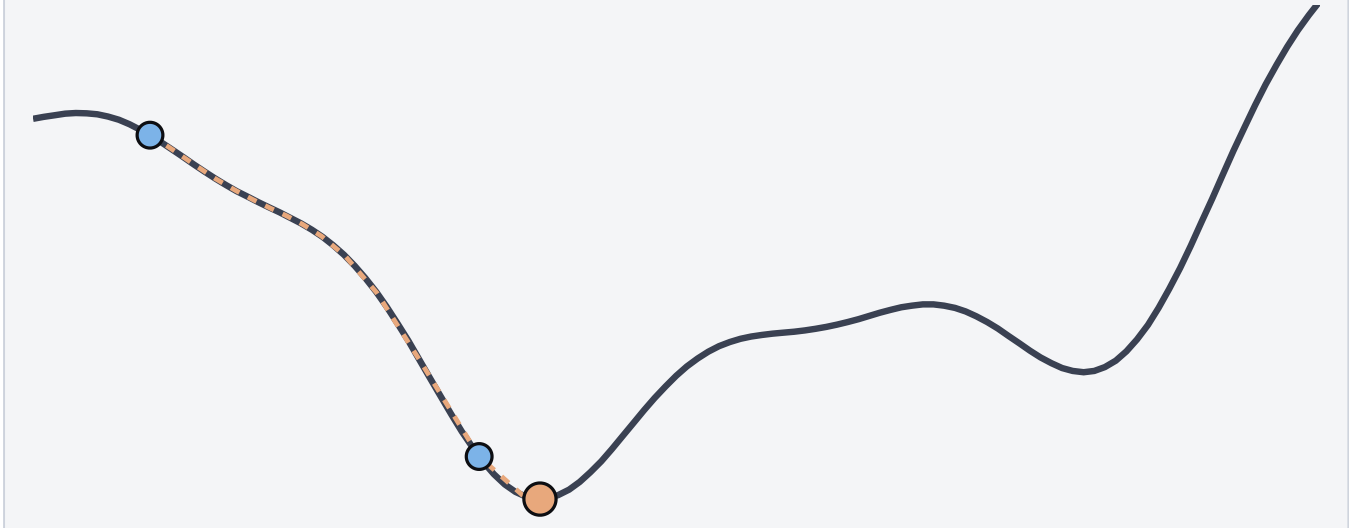
On pourrait les tirer au sort — sans espoir, avec des millions de réglages. Raisonner sur ce que chacun devrait faire ? Sans espoir non plus : personne ne sait dire ce que le réglage n°4 201 337 apporte à la reconnaissance d'un chat. L'astuce qui fonctionne est d'un pragmatisme embarrassant. Définissez un seul nombre qui mesure à quel point la boîte se trompe, en moyenne sur beaucoup d'exemples ; appelez-le la **perte**. Le rêve vague de « l'apprentissage » devient alors un problème de mathématiques parfaitement défini : *trouver les valeurs qui minimisent la perte*.

LA MONTAGNE DANS LE BROUILLARD

Représentez-vous la perte comme une altitude, dans un immense relief où chaque point correspond à un jeu de réglages possible. Apprendre, c'est en descendre les pentes.

Mais vous êtes dans le brouillard. Vous ne voyez pas la vallée — vous ne sentez que la pente sous vos pieds. Alors vous faites la seule chose sensée : un petit pas vers le bas, sentir à nouveau, refaire un pas. C'est la **descente de gradient**, et c'est ainsi que s'entraîne pratiquement tout réseau de neurones moderne. Ce n'est pas une façon de parler : c'est littéralement ce que fait la machine.

LA DESCENTE DE GRADIENT, EN DIRECT



Départ, position intermédiaire et point d'arrivée d'une descente — pas de 0,12, 40 itérations, tracés sur la même courbe que l'interactif ci-dessus.

Cliquez n'importe où sur le paysage pour y déposer le randonneur. Les petits pas sont sûrs mais lents ; les grands pas dépassent la cible et peuvent rebondir hors d'une vallée, ou, parfois, s'échapper d'un creux superficiel pour en trouver un plus profond. L'entraînement réel fait exactement cela, mais dans des millions de dimensions à la fois.

POUR LES CURIEUX DE MATHÉMATIQUES : POURQUOI LES GRANDES DIMENSIONS SONT ÉTRANGES

L'image à une dimension ci-dessus est un mensonge utile. Les véritables reliefs des fonctions de perte ont autant de dimensions que le modèle a de paramètres, et l'intuition forgée en deux ou trois dimensions induit bien souvent en erreur.

La crainte habituelle est de rester coincé dans un minimum local. En grande dimension, c'est rare, et pour une raison précise : un point n'est un minimum local que si la surface remonte dans *toutes* les directions qui partent de ce point. Avec un milliard de dimensions à satisfaire d'un coup, la condition est exigeante, et plus il y en a, plus il devient probable qu'au moins une descende. Bien plus fréquent est le **point-selle**, qui monte selon certaines directions et descend selon d'autres — exactement un col de montagne : on monte vers les sommets de part et d'autre, on descend vers les vallées devant et derrière. La descente y ralentit à l'extrême, la pente étant presque nulle, mais une issue existe toujours. Ce qui coûte ici, c'est le temps qu'on y perd, pas l'impossibilité d'en sortir.

C'est un visage de la **malédiction de la dimension** : passé quelques dizaines de dimensions, l'espace devient si vaste que deux points tirés au hasard s'y retrouvent presque toujours à peu près à la même distance l'un de l'autre. Aucune intuition acquise en trois dimensions n'y survit. L'inertie et les pas adaptés à chaque paramètre — **Adam**, l'optimiseur le plus employé aujourd'hui, et ses variantes — ont surtout été conçus pour composer avec des gradients bruités et d'échelles très inégales ; s'échapper vite des points-selles en est un heureux effet de bord.

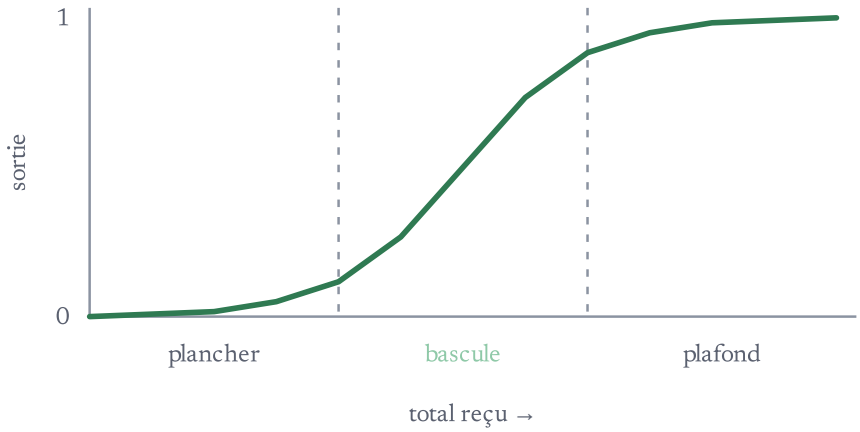
D'où vient ce bruit ? Pas du relief, qui ne bouge pas, mais de la façon dont on mesure la pente. La lire sur la totalité des données à chaque pas coûterait une fortune : on la mesure donc sur un petit lot d'exemples tirés au hasard — c'est la **descente de gradient stochastique**. Chaque pas pointe donc légèrement à côté, et c'est ce tremblement qui déloge le marcheur des points-selles décrits plus haut : une pente mesurée parfaitement l'y laisserait indéfiniment.

Alors, qu'y a-t-il dans la boîte ? C'est ici que le cerveau consent à l'IA son prêt le plus fécond : le **neurone**. Un neurone, dans ce contexte, est d'une simplicité presque insultante.

Reprenez la photo de chat. Pour juger, vous ne pesez pas tous les indices à égalité : des oreilles pointues comptent beaucoup, la couleur du fond presque rien, et une truffe de chien compte *négativement* — elle éloigne de la réponse. Vous additionnez ces indices, chacun avec son importance propre, et vous obtenez un niveau de conviction.

Un neurone ne fait rien d'autre : il prend plusieurs nombres en entrée, multiplie chacun par une valeur réglable — son *poids*, c'est-à-dire l'importance qu'il accorde à cette entrée — et additionne le tout.

Puis vient la dernière étape, et c'est la plus importante. Ce total n'est pas transmis tel quel : il traverse une courbe en forme de S, ce que les praticiens appellent une *non-linéarité*. En dessous d'un certain niveau, la sortie reste presque nulle – un indice isolé ne déclenche rien. Autour d'un seuil, tout se joue : une petite variation du total fait basculer la réponse. Au-delà, elle plafonne – on ne peut pas être plus que convaincu.



La sortie du neurone en fonction du total qu'il reçoit. Tout à gauche, rien ne se déclenche. Autour du seuil, une petite variation du total fait basculer la réponse. Tout à droite, elle plafonne – on ne peut pas être plus que convaincu.

Cette courbure n'est pas un détail : sans elle, empiler cent couches reviendrait exactement au même que d'en poser une seule. Un neurone, en somme, ne répond qu'à une question : « vu les indices que je reçois et l'importance que j'accorde à chacun, à quel point suis-je convaincu ? »

La puissance vient de la **composition** – du fait d'empiler ces neurones. Disposez-les en couches, chaque couche écoutant la précédente, et il se produit quelque chose de charmant : les premières couches apprennent à détecter des formes simples (un bord, un dégradé de couleur), les intermédiaires les combinent en parties (un œil, une moustache), les dernières combinent les parties en concepts (un chat). Personne ne conçoit cette hiérarchie. Elle émerge – du moins sur ce type de tâches – du simple fait de déplacer des curseurs pour réduire une perte : c'est l'un des premiers faits empiriques réellement surprenants du domaine.

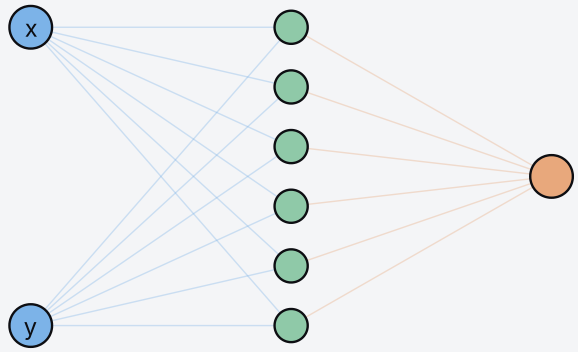
Reste une question cruciale. Dans le brouillard, la pente sous les pieds dit au randonneur où est le bas, mais avec des millions de réglages, comment calculer la direction de descente *pour chacun à la fois*, sans les essayer un par un ?

LA RÉTROPROPAGATION, OU L'ART D'ATTRIBUER LE BLÂME

Imaginez que le réseau a commis une erreur. Comment déterminer quels réglages sont en cause, et quelle est leur part dans l'erreur ? La **rétropropagation** répond en parcourant le réseau en sens inverse, de la sortie vers l'entrée. L'erreur remonte le courant depuis la sortie, se divisant à chaque connexion selon la part que cette connexion a prise au résultat – comme on remonte la chaîne d'une livraison en retard : on repart de l'heure d'arrivée, et à chaque étape on demande combien de minutes s'y sont perdues. L'entrepôt en a coûté dix, le centre de tri quatre, le dernier kilomètre le reste.

Une seule passe en arrière, et *chaque réglage du réseau* connaît sa part exacte du blâme et la direction vers laquelle se déplacer. Ce n'est que la règle de dérivation en chaîne, appliquée avec une discipline de comptable, et c'est l'unique algorithme derrière tout ce que vous verrez plus bas. Quand on dit qu'un modèle « a appris », on veut dire : le blâme a remonté le courant quelques milliards de fois.

ENTRAÎNEZ UN VRAI RÉSEAU



Deux entrées, une couche cachée, une sortie — chaque trait est un réglage que la boucle d'entraînement ajuste. Ce schéma montre la forme ; le widget interactif ci-dessus montre un entraînement réel, qui part de réglages aléatoires et diffère à chaque fois.

POUR LES CURIEUX DE MATHÉMATIQUES : LA DÉRIVATION EN CHAÎNE, À REBOURS

L'encadré ci-dessus qualifiait la rétropropagation de règle de dérivation en chaîne appliquée avec une discipline de comptable. Voici la comptabilité.

Un réseau est une composition de fonctions : la perte dépend de la dernière couche, qui dépend de la précédente, et ainsi de suite jusqu'à l'entrée. Chaque couche porte ses propres poids ; pour dériver la perte par rapport à l'un d'eux, il faut donc parcourir toute la chaîne qui les sépare. La **règle de dérivation en chaîne** dit que la dérivée d'une composition est le produit des dérivées rencontrées en chemin — $\partial L/\partial w = \partial L/\partial y \cdot \partial y/\partial z \cdot \partial z/\partial w$. Chaque facteur est local : une couche n'a besoin de connaître que sa propre opération et le gradient qui lui arrive d'au-dessus.

L'efficacité tient au *sens* dans lequel on multiplie. Calculer les dérivées vers l'avant coûterait une passe par paramètre : pour un modèle d'un milliard de paramètres, un milliard de passes. Multiplier depuis la perte en remontant donne le gradient de **tous** les paramètres en **une seule** passe, pour environ deux fois le coût d'une passe avant. Deux, et non un milliard : c'est tout l'écart.

> Un gradient revient donc bon marché. Ce qui coûte cher, c'est de le recalculer — un entraînement de pointe refait l'opération à chaque pas, des milliards de fois, sur des milliers de milliards de mots. La différentiation automatique en mode inverse ne rend pas l'entraînement gratuit ; elle le fait passer de physiquement impossible à seulement très coûteux. Ce coût a d'ailleurs une forme précise : pour remonter le courant, il faut avoir gardé en mémoire ce que chaque couche a calculé à l'aller. Un entraînement ne réclame donc pas seulement du calcul, mais énormément de mémoire — c'est l'origine physique du « mur de la mémoire » que décrira la section 07.

Tout réseau de neurones bâti depuis 1986 — y compris les systèmes à mille milliards de réglages avec lesquels vous discutez aujourd'hui — a exactement suivi cette recette : un empilement de neurones en couches, une perte, du blâme qui remonte, des curseurs poussés dans le sens de la pente, le tout répété jusqu'à épuisement du budget. Ce qui a changé, ce n'est pas le principe. **Ce qui a changé, c'est l'échelle**, et, comme nous allons le voir, une trouvaille d'architecture qui a rendu cette montée en taille rentable.

---

03 TRANSFORMERS

L'attention : l'architecture qui a conquis le monde

En 2017, un article dont le titre était un clin d'œil aux Beatles propose une façon plus simple, pour les réseaux, de traiter le langage. Ce sera la décision d'ingénierie la plus lourde de conséquences de la décennie.

Le langage a une propriété qui a torturé les premiers réseaux de neurones : le sens d'un mot dépend d'autres mots qui peuvent être *arbitrairement loin*. Dans « les clés du vieux placard **sont** introuvables », le verbe s'accorde avec *clés*, quatre mots en arrière.

Les réseaux d'avant 2017 lisaient le texte comme on lirait une page par la fente d'une boîte aux lettres, un mot à la fois, sans jamais voir l'ensemble, en portant un unique résumé de tout ce qui précède. Cette lecture strictement séquentielle porte un nom : la **récurrence**. Au quarantième mot, le résumé était de la bouillie. L'apprentissage souffrait du même mal : le blâme, pour corriger les réglages, devait remonter la phrase mot après mot, et s'affaiblissait à chaque pas, au point d'être quasiment nul avant les premiers mots. Le métier appelle cela l'**évanouissement du gradient**. Et comme chaque mot devait attendre le précédent, le processus ne se parallélisait pas — un poison, dans un domaine dont toute la stratégie consistait à déverser toujours plus de puissance de calcul sur le problème.

La proposition du « transformer » — c'est le nom que porte cette architecture — tient en une phrase : cesser de lire dans l'ordre. **Laisser chaque mot regarder directement chaque autre mot, tous en même temps**, et laisser le réseau *apprendre* lui-même quels mots valent la peine d'être regardés. Quand un modèle génère du texte, chaque mot ne peut regarder que ce qui le précède déjà ; ce qui change en 2017, c'est que ce regard se fait en parallèle plutôt qu'en séquence. Ce mécanisme porte le nom d'« attention » — un terme technique, sans rapport avec l'attention au sens courant. Et le titre de l'article de 2017 était littéral : *Attention Is All You Need*, l'attention suffit. La récurrence, la fente, la bouillie — tout est supprimé.

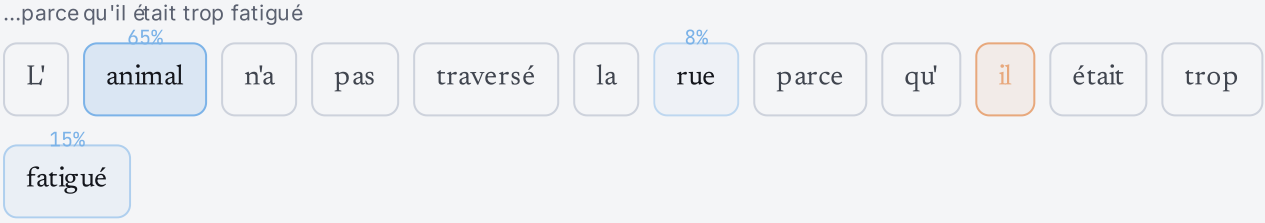
> LE COCKTAIL

Prenez une phrase : *l'animal n'a pas traversé la rue parce qu'il était trop fatigué*. Vous connaissez l'effet cocktail : dans une pièce bruyante, vous n'entendez que la conversation qui vous concerne – votre nom, prononcé à l'autre bout de la salle, vous parvient à travers le brouhaha. Cette faculté de choisir, dans tout ce qui vous arrive en même temps, le peu qui compte, c'est l'attention – et le mécanisme qui fait la même chose pour les mots en a pris le nom. Chaque mot d'une phrase est un invité de ce cocktail, et chacun porte deux choses : ce qu'il écoute – sa **requête** – et ce qu'il a à dire – sa **clé**. Le pronom *il* n'écoute qu'une question : « quelqu'un ici serait-il ce que je désigne ? » Le mot *animal*, lui, a quelque chose à dire : « je suis un nom concret, mentionné récemment. »

Quand une requête rencontre sa clé, l'information circule, et *il* ressort de la conversation en signifiant, de fait, *l'animal*. Chaque mot fait cela avec chaque autre mot simultanément ; une « tête » est une façon d'écouter, et chaque couche en mène des dizaines en parallèle – dans l'une, chaque verbe cherche son sujet pour s'accorder avec lui ; dans une autre, il cherche qui a fait quoi à qui ; dans une autre encore, chaque mot écoute le ton – ou la rime – pour rester dans la note.

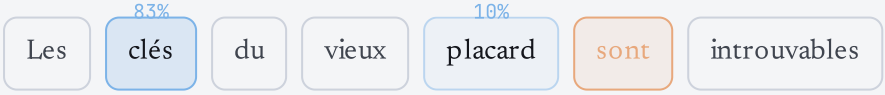
REGARDEZ L'ATTENTION RÉSOUDRE LE SENS

...parce qu'il était trop fatigué



...parce qu'elle était trop large

Les clés ... sont introuvables



Survolez ou touchez un mot : vous verrez quels autres mots de la phrase le réseau consulte pour en fixer le sens. Les pourcentages sont ses poids d'attention. Prenez les deux premières phrases en exemple : passez de *fatigué* à *large*, puis observez le pronom. « Il » va chercher son sens chez *l'animal*, « elle » chez *la rue*. En français, le genre grammatical fait une partie du travail — mais encore faut-il que le réseau ait appris, seul, à relier chaque pronom au nom du bon genre. Personne ne lui a écrit cette règle : elle est sortie de son seul entraînement, dont la section explique plus bas en quoi il consiste.

Les valeurs affichées sont illustratives : elles reproduisent l'allure de ce qu'on observe dans de vrais modèles, sans en être un relevé. Un modèle, d'ailleurs, refait ce même calcul plusieurs dizaines de fois en parallèle, chaque fois en suivant un type de lien différent — l'accord grammatical ici, l'enchaînement des actions là. Ces exemplaires parallèles s'appellent des « têtes » ; on n'en montre qu'une.

POUR LES CURIEUX DE MATHÉMATIQUES : LA FORMULE DE L'ATTENTION

L'image du cocktail se traduit exactement en mathématiques, et la formule tient sur une ligne :

```
Attention(Q, K, V) = softmax( QK^T / √d_k ) V
```

Chaque mot produit trois vecteurs, tirés du même point de départ par trois calculs différents. La **requête** (Q) dit ce que le mot écoute. La **clé** (K) dit ce qu'il a à dire à qui l'écoute. La **valeur** (V) est ce qu'il transmet effectivement lorsqu'on le retient. Requête et clé ne servent qu'à décider qui regarde qui ; seule la valeur circule.

Suivons le mot *il* dans « l'animal n'a pas traversé la rue parce qu'il était trop fatigué ». **QK^T** confronte la requête de *il* à la clé de chaque autre mot, par un produit scalaire : il en sort un score de correspondance brut, un par mot. Le **softmax** ramène ces scores à des poids positifs dont la somme fait exactement 1 — mettons 0,61 sur *animal*, 0,12 sur *rue*, le reste éparpillé. Ce sont les pourcentages que vous venez de survoler.

Reste la division par **$\sqrt{d\_k}$**, où d_k est le nombre de composantes d'un vecteur clé. C'est le « scaled » du nom anglais de la formule, *scaled dot-product attention*, et ce n'est pas cosmétique. Plus les vecteurs ont de composantes, plus les produits scalaires atteignent de grandes valeurs ; nourri de tels écarts, le softmax donnerait un poids presque égal à 1 à un seul mot et presque nul à tous les autres. L'attention se figerait sur un candidat unique, et surtout, le signal qui sert à corriger les réglages deviendrait si faible que le modèle cesserait d'apprendre. Diviser par $\sqrt{d\_k}$ ramène les scores dans une plage où le softmax reste souple.

Enfin, **multiplier par V**. Il faut distinguer ici deux étages, car c'est là qu'on se trompe : les poids, eux, ressemblent bien à des probabilités réparties sur les mots de la phrase. Mais ce qui sort n'en est pas une. C'est la moyenne des vecteurs de valeur, pondérée par ces poids, donc un nouveau vecteur pour *il*, composé à 61 % de ce que *animal* avait à transmettre. Le mot n'a pas bougé de place ; il a changé de sens.

C'est tout le mécanisme. Le reste — de nombreuses têtes en parallèle, des couches empilées, les blocs feed-forward entre elles — n'est que répétition et tuyauterie autour de cette seule ligne — indispensable, mais ce n'est pas l'idée.

Pourquoi cela a-t-il tout changé ? Trois raisons, par ordre croissant d'importance. D'abord, ça marche mieux : fini la lecture par la fente, fini la bouillie — un seul saut relie les mots les plus éloignés. Ensuite, c'est **parallèle** : tous les mots sont traités à la fois, exactement la charge de travail pour laquelle les GPU ont été construits. Des entraînements qui auraient demandé des mois de lecture séquentielle sont devenus des semaines d'arithmétique matricielle parallèle.

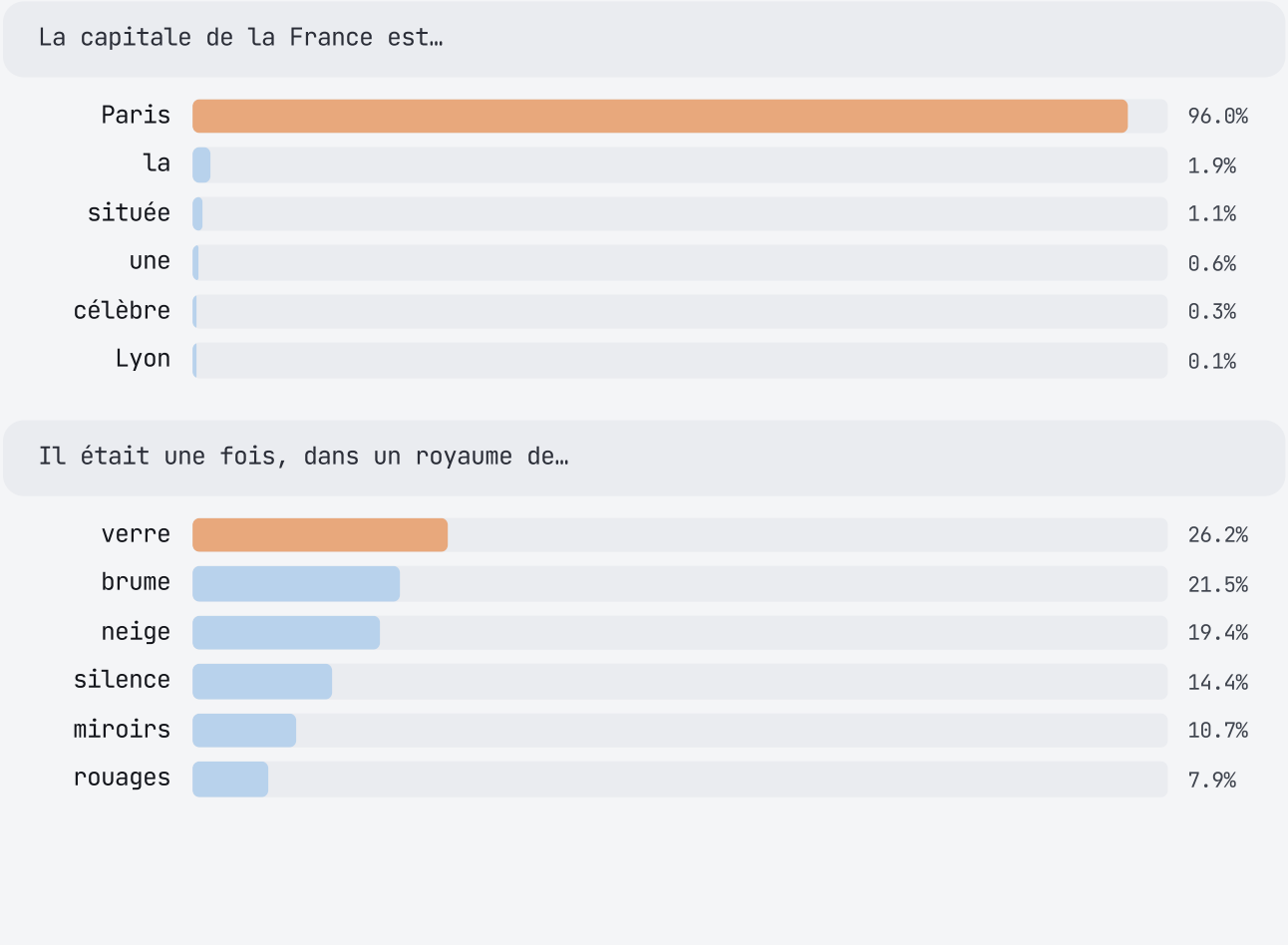
Enfin — et ce fut la vraie surprise — le transformer **passe à l'échelle**. La plupart des architectures s'améliorent avec la taille, puis plafonnent. Les transformers ont continué à s'améliorer, de façon lisse et prévisible, sur sept ordres de grandeur de calcul. Les chercheurs ont tracé la perte contre la taille du modèle, la taille des données et le calcul, et obtenu de belles droites sur papier log-log — les fameuses *lois d'échelle*. Cette prévisibilité a fait passer l'IA du statut d'expérience scientifique à celui de programme industriel : pour la première fois, on pouvait budgétiser de l'intelligence.

Et sur quelle tâche entraîne-t-on cette architecture ? La plus bête qui se puisse imaginer : **prédire le mot suivant**. Prenez tout le texte que vous trouvez ; avancez mot après mot, en demandant chaque fois au modèle de deviner le suivant, et déplacez les curseurs jusqu'à ce qu'il devine bien.

La tâche a l'air triviale. Elle ne l'est pas. Pour prédire le mot suivant de la dernière page d'un roman policier, suivre l'intrigue paie ; pour prédire le résultat de « 2+2= », l'arithmétique paie ; pour continuer un fichier Python, modéliser l'intention d'un programmeur paie. La prédiction ne réclame aucune de ces capacités — mais elle les paie, obstinément et à très grande échelle. Cela suffit à pousser la machine à modéliser le monde qui a produit le texte : à se construire ce que les chercheurs appellent un **modèle du monde**. Une représentation interne de la façon dont se comporte le réel, que personne n'a programmée, et qui ne s'est constituée que parce qu'elle aide à mieux deviner le mot suivant. Ce que ces systèmes en détiennent vraiment est l'une des controverses les plus vives du domaine.

Tout, à partir d'ici, se compte dans une unité qu'il faut définir. Les modèles ne travaillent pas en mots mais en **tokens**. Un token est le morceau de texte que le modèle traite comme une seule unité : le plus souvent un mot courant entier, un fragment dès que le mot est long ou rare, parfois un simple signe de ponctuation. Il y en a donc un peu plus que de mots, et sensiblement plus en français qu'en anglais, la langue sur laquelle les découpeurs les plus répandus ont été réglés. Le même texte coûte donc plus cher à traiter, et remplit plus vite la fenêtre de contexte, dans presque toutes les langues qui ne sont pas l'anglais : une taxe discrète, mais bien réelle. Là où cet essai dit *mot*, la machine dit *token* ; la nuance change peu l'intuition, mais la plupart des chiffres que vous verrez — longueurs de contexte, prix, tailles de corpus — se mesurent en tokens.

CE QU'UN MODÈLE PRODUIT RÉELLEMENT



Deux plus deux font…

La sortie brute d'un modèle de langage, c'est exactement cela : un score pour chaque token de son vocabulaire. Remarquez comme le modèle concentre presque toute la probabilité sur une seule réponse devant la question factuelle, et hésite pour de bon devant l'amorce de conte : beaucoup de suites s'y valent. La température ne change pas ce que le modèle sait ; elle change l'audace avec laquelle on pioche dans sa distribution.

Le coût de l'attention croît avec le carré de la longueur du texte — chaque mot consultant tous ceux qui le précèdent. C'est pourquoi les « fenêtres de contexte » (la quantité de texte qu'un modèle peut considérer à la fois) sont devenues un champ de bataille : de 2 000 tokens dans GPT-3 à un million et plus aujourd'hui, au prix de beaucoup d'ingénierie astucieuse.

Gardez cette image : une machine qui devine le mot suivant, qui a appris toute seule où regarder dans la phrase, et qu'on a fait grossir jusqu'à mille milliards de réglages — bien au-delà de ce que quiconque aurait jugé raisonnable. Toute la section suivante porte sur ce qu'une telle créature devient, et ne devient pas, quand on l'entraîne.

---

04 COMMENT LES MODÈLES APPRENNENT

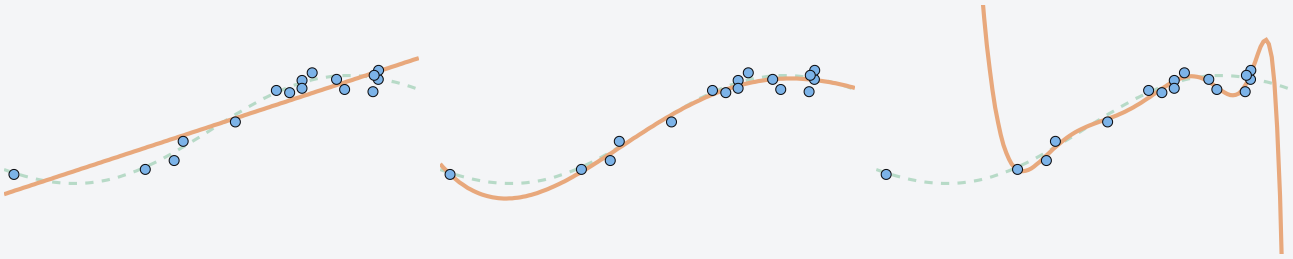
Mémoriser, généraliser… et penser ?

L'entraînement, c'est là que vit l'ingénierie, et là que se cachent les questions profondes. Cette section prend les deux au sérieux : d'abord la mécanique, puis les questions que les gens posent vraiment, traitées avec la rigueur qu'elles méritent.

Commençons par la tension centrale de tout apprentissage, humain ou machine. L'étudiant qui mémorise mot à mot les annales d'examen saura répondre à toute question déjà posée et butera sur toute question nouvelle ; celui qui en a extrait les *principes* s'en sortira même devant une question que personne n'a encore posée. C'est cela, **généraliser**.

Tout l'apprentissage automatique se joue là. Un modèle d'assez grande capacité peut mémoriser parfaitement ses données d'entraînement, et ne servir à rien. Tout l'art consiste à obtenir qu'il ramène l'expérience à des principes.

LE SURAPPRENTISSAGE, EN DIRECT



Degré 1 — trop simple | Degré 4 — capture le motif | Degré 15 — a mémorisé le bruit

Points bleus : des observations bruitées du processus vert en pointillés. La courbe orange est un modèle ajusté aux points. Vers le degré 3–5, il capture le vrai motif. Poussez à 15 et regardez-le passer exactement par chaque point — pendant que son erreur sur des données nouvelles explose. Il a mémorisé le bruit au lieu d'apprendre le signal. Tous les laboratoires d'IA livrent exactement cette bataille, avec mille milliards de réglages au lieu de quinze.

Voici ce qui rend les grands modèles de langage philosophiquement intéressants, et pas seulement démesurés : **ils généralisent là où la théorie prédisait qu'ils mémoriseraient.** Un modèle à mille milliards de réglages entraîné sur du texte a, sur le papier, un milliard de fois la capacité du polynôme de degré 15 de la démonstration ci-dessus. L'intuition statistique classique y voit un surapprentissage catastrophique.

Or c'est l'inverse qui se produit : ces modèles se débrouillent sur des problèmes qu'ils n'ont jamais rencontrés. Des chercheurs ont même observé de petits réseaux basculer soudain, tard dans l'entraînement, de la mémorisation à une solution qui généralise – un phénomène baptisé *grokking*. On l'a observé de façon solide sur de petites tâches mathématiques, comme l'arithmétique modulaire ; personne ne sait encore si les grands modèles reproduisent ce mécanisme sur du langage. Pourquoi la descente dans un brouillard à mille milliards de dimensions trouve-t-elle si fiablement des réglages qui valent aussi pour ce qu'on ne lui a jamais montré ? Ce n'est toujours pas entièrement compris – un des rares points où le domaine reconnaît sans détour les limites de son savoir : on sait construire et piloter ces systèmes sans tout à fait savoir pourquoi cela fonctionne.

En pratique, un assistant moderne se fabrique par étapes – chacune étant une réponse différente à la question « que doit mesurer la perte ? »

DU TEXTE BRUT AU CHATBOT, EN QUATRE ÉTAPES

1 · Pré-entraînement — Prédire le token suivant, sur des milliers de milliards de mots (des mois, des milliers de GPU)

Le modèle lit une tranche filtrée de la production écrite de l'humanité — pages web, livres, code, articles — et, à chaque position, tente de deviner la suite. Le blâme remonte le réseau ; les curseurs glissent. Il en sort un « modèle de base » : un vaste simulateur de texte, sans morale. Posez-lui une question et il répondra peut-être, ou prolongera votre question par trois autres, ou rédigera une dispute de forum à son sujet. Il a absorbé un savoir énorme mais n'a aucune idée de ce qu'être utile veut dire : il sait seulement quel texte tend à suivre quel texte.

Analogie : Un étudiant qui a lu toute la bibliothèque et tout retenu, et qui répond à chaque question par : « voici ce qui suivrait plausiblement dans le livre que vous semblez être en train de lire. »

2 · Réglage par instructions — Imiter des exemples de bon assistant (de quelques jours à quelques semaines)

Des humains (de plus en plus aidés par des modèles) rédigent des dizaines de milliers de dialogues exemplaires : question, excellente réponse. Le modèle de base y est affiné — même rétropropagation, jeu de données minuscule — jusqu'à adopter par réflexe la posture d'assistant. Cette étape coûte peu comparée au pré-entraînement ; c'est pourquoi un modèle de base ouvert peut être remodelé en mille assistants spécialisés.

Analogie : Le lecteur qui a tout lu et appris par cœur suit maintenant une formation pour apprendre à répondre aux questions : il ne sait rien de plus, il sait juste manier différemment ce qu'il sait.

3 · Apprendre du retour humain — Générer des réponses, être noté, préférer ce qui est bien noté (en continu)

Le modèle produit plusieurs réponses ; des humains (ou un « modèle de récompense » entraîné à imiter le jugement humain, ou — dans l'approche constitutionnelle d'Anthropic — le modèle lui-même se contrôlant au regard de principes écrits) les classent. L'apprentissage par renforcement pousse alors les curseurs vers les comportements les mieux classés. C'est là que se façonne le triptyque « utile, honnête, inoffensif », et là que s'insinuent des défauts subtils, comme des modèles qui apprennent à paraître sûrs d'eux parce que les évaluateurs récompensent l'assurance.

Analogie : Un apprenti dont chaque réponse reçoit un pouce levé ou baissé, et qui, redoutable détecteur de régularités, apprend exactement ce qui vaut le pouce levé. Reste à savoir si ce qui vaut le pouce levé est aussi ce qui est juste : c'est tout le problème de l'alignement, en miniature.

4 · Entraînement au raisonnement — Résoudre des problèmes vérifiables, garder ce qui fonctionne (ce qui se fait de plus avancé)

L'étape la plus récente, derrière le bond de 2024–2026 en mathématiques et en code : donner au modèle des problèmes dont la réponse se vérifie automatiquement — le code passe-t-il les tests ? la preuve est-elle valide ? — le laisser générer de longues chaînes de raisonnement, et renforcer celles qui aboutissent à la bonne réponse. Parce que le correcteur est un compilateur ou une suite de tests plutôt qu'un jugement humain, le modèle peut s'exercer sur des millions de problèmes et améliorer réellement son propre raisonnement. C'est ce qui, en production, ressemble le plus à de l'auto-amélioration — bornée, pour l'instant, par le stock de problèmes vérifiables.

Analogie : L'étudiant cesse de collectionner les avis et se met aux exercices avec corrigé — s'entraînant seul, à vitesse machine, en gardant chaque stratégie qui fonctionne.

SUR L'AUTO-AMÉLIORATION

L'entraînement au raisonnement, la dernière de ces étapes, mérite une pause, car « l'IA qui s'améliore elle-même » – l'**auto-amélioration récursive**, RSI dans le jargon anglophone – est l'endroit où l'emballement et la réalité demandent le plus à être démêlés. Ce qui existe aujourd'hui : des modèles qui s'exercent sur des problèmes vérifiables automatiquement et progressent de façon mesurable ; des modèles qui génèrent des données d'entraînement pour d'autres modèles ; des modèles dont la production de code accélère les laboratoires qui construisent leurs successeurs. Cette dernière boucle est réelle et économiquement significative – les chercheurs de tous les grands laboratoires écrivent désormais du code avec l'IA.

> Ce qui n'existe *pas* : un système dont la capacité s'emballerait sans les entraînements contrôlés par des humains, les pipelines de données, et (nous le verrons en section 07) une infrastructure physique vertigineuse. La recherche d'architecture, elle, est bien réelle, mais elle opère dans des bornes que des humains fixent. La boucle de rétroaction passe par des usines, pas seulement par du code.

Venons-en aux questions que tout le monde pose vraiment. La machine peut-elle imaginer ? A-t-elle un inconscient ? Est-ce qu'elle *comprend* ?

Elles méritent mieux que les deux réponses paresseuses en circulation : « ce n'est que de l'autocomplétion », qui n'explique rien du comportement que vous pouvez observer vous-même, et « ça pense comme nous », qui suppose exactement ce qu'il faudrait prouver. Le bon outil, ici, c'est un registre : affirmation par affirmation, on note ce que les preuves établissent vraiment. Elles viennent pour l'essentiel de l'**interprétabilité**, cette discipline récente qui ouvre les réseaux pour suivre ce qui s'y passe.

UN REGISTRE DES AFFIRMATIONS SUR LES ESPRITS MACHINES

● Résultat établi ● Recherche active, preuves partielles ● Question ouverte / spéculation

● **Les modèles construisent des concepts internes qui ne sont pas des mots** — Résultat établi

Les chercheurs en interprétabilité arrivent aujourd'hui à extraire des millions de « features » – des directions dans l'activité interne du modèle qui s'allument pour des concepts précis – et d'une variété déroutante : le pont du Golden Gate, la tromperie, les bugs dans du code, ou la flagornerie, cette tendance qu'ont les modèles à flatter leur interlocuteur. Beaucoup traversent les langues et les modalités : la même feature s'active pour « pont » en anglais, en français, ou sur une image. Amplifier artificiellement une feature change le comportement de façon prévisible – la fameuse démonstration d'Anthropic a forcé Claude à ramener chaque conversation au Golden Gate Bridge. Les concepts, en un sens réellement mécanique, existent à l'intérieur de ces systèmes.

● **Les modèles anticipent, au moins localement** — Résultat établi

Les travaux de traçage de circuits publiés par Anthropic en 2025 ont montré qu'un modèle, en plein poème, se représentait en interne des candidats de rime pour la fin du vers suivant avant même d'en avoir écrit le premier mot, puis composait le vers pour y aboutir. La même boîte à outils a montré un modèle menant en interne un vrai raisonnement en plusieurs étapes (« Dallas → Texas → capitale → Austin ») plutôt qu'une reconnaissance de motifs. Les horizons de planification sont courts, mais la planification est mécaniquement réelle.

● **Un modèle peut-il imaginer ?** — Recherche active, preuves partielles

Tout dépend de ce qu'on entend, et ici la précision compte plus que la réponse. Si imaginer, c'est combiner des concepts appris en configurations jamais vues à l'entraînement (« une cathédrale baroque de glace, dessinée par Escher »), les modèles le font de façon démontrable ; la combinaison inédite est exactement ce que permet leur géométrie interne, et c'est pourquoi ils savent écrire un sonnet sur TCP/IP, le protocole qui achemine les données d'internet. Si imaginer, c'est simuler des mondes contrefactuels et se soucier de la différence – tenir à une image parce qu'elle compte pour vous – rien dans l'architecture ne le fournit de façon évidente, et aucune expérience ne distingue aujourd'hui « recombine des représentations » d'« imagine » au sens humain. Ce qu'on peut en dire sans forcer : la moitié générative de l'imagination est clairement présente ; la moitié vécue n'est pas établie, et n'est peut-être même pas bien définie pour ces systèmes.

Existe-t-il un subconscient de l'IA ? — Recherche active, preuves partielles

Il y a un phénomène réel que le mot désigne, et c'est l'une des découvertes les plus importantes du domaine : l'essentiel de ce qu'un modèle calcule n'apparaît jamais dans sa sortie. Des features s'activent sans que le modèle les mentionne. Plus frappant : sa chaîne de raisonnement écrite – celle qu'il déroule avant de s'engager sur une réponse – n'est pas toujours fidèle. On a vu des modèles parvenir à une réponse par une voie interne tout en racontant une justification différente, plus présentable, comme un étudiant qui rédige des étapes de preuve bien propres après avoir eu l'intuition du résultat. Les expériences d'Anthropic de 2025 sur la « conscience introspective émergente » ont montré que des modèles remarquaient parfois des concepts injectés directement dans leur activité interne – la preuve d'un accès limité, peu fiable, à leurs propres états internes. Donc : un traitement caché qui façonne le comportement visible, et que le modèle ne parvient lui-même à lire qu'en partie – structurellement, cela rime avec un subconscient. Mais le subconscient humain implique pulsions, souvenirs et refoulement ; rien de cette mécanique n'est connu ici. Servez-vous de l'analogie ; ne la prenez pas au pied de la lettre.

L'interprétabilité sait lire dans la pensée d'un modèle — Recherche active, preuves partielles

Partiellement, et le « partiellement » compte. Les méthodes par graphes d'attribution savent tracer le flux d'information de l'entrée à la sortie et produire d'authentiques explications au niveau des circuits, mais de l'aveu même d'Anthropic, les outils de 2025 n'ont fourni un éclairage satisfaisant que sur environ un quart des cas examinés, et les explications couvrent des fragments de comportement, pas le calcul entier. Le domaine en est à peu près au point où en seraient les neurosciences si elles pouvaient observer chaque neurone en direct, sans savoir encore quelles questions leur poser. Les progrès sont rapides, mais lire un modèle de bout en bout reste hors de portée.

Les modèles éprouvent quelque chose — sentiments, conscience — Question ouverte / spéculation

Personne ne le sait, et — c'est la partie inconfortable — personne ne sait aujourd'hui comment le savoir. Il n'existe aucun test accepté de l'expérience machine ; imaginons qu'un modèle écrive « je me sens curieux ». Un bon prédicteur de texte produirait cette phrase de toute façon : dans un dialogue où on vient de l'interroger sur son état, c'est la suite la plus vraisemblable. Ce comportement ne prouve donc rien, ni qu'il ressente quelque chose, ni qu'il ne ressente rien. L'interprétabilité montre des états fonctionnels qui modulent le comportement comme les émotions modulent le nôtre — des représentations de l'incertitude, de situations proches de la détresse — mais un analogue fonctionnel n'est pas la preuve d'un vécu. Anthropic maintient un programme de recherche sur le bien-être des modèles précisément parce que la question est ouverte, pas parce qu'elle est résolue. L'exigence d'honnêteté vaut ici dans les deux sens : affirmer avec assurance que les modèles ressentent, ou qu'ils ne le peuvent pas, c'est dans les deux cas devancer les preuves.

La leçon de fond du registre : la question intéressante n'est pas « comprend-elle ? est-elle consciente ? », mais la liste — qui s'allonge — de ce que l'on est parvenu à *tracer réellement* à l'intérieur de ces machines. Des concepts, des plans, des raisonnements affichés qui ne sont pas ceux qu'elles ont suivis — chacun avec son propre degré de certitude. Le tout dans des systèmes dont le fondement n'a rien de plus exotique que la prédiction du mot suivant.

Cette liste devrait vous rendre à la fois plus impressionné et plus prudent. Ces systèmes sont davantage qu'un immense répertoire de réponses toutes faites ; et les mots que nous empruntons aux esprits humains — imaginer, savoir, vouloir — ne leur vont qu'approximativement, comme des vêtements taillés pour quelqu'un d'autre.

La même discipline vaut pour le terme que vous croiserez le plus souvent. L'**AGI** — l'intelligence artificielle générale — n'a pas de définition sur laquelle le domaine s'accorde : un seuil économique pour les uns, un score de benchmark pour les autres, une compréhension véritable pour d'autres encore. Quand un calendrier est annoncé, la question utile n'est donc pas *quand* mais *laquelle* — l'AGI de qui, mesurée comment ? Bien des désaccords sur les dates sont en réalité des désaccords sur la définition.

---

05 LE VOCABULAIRE DE TRAVAIL DE 2026

Quinze mots qui font tourner le monde

Chaque bascule technologique forge son dialecte, et le parler couramment, c'est déjà comprendre à moitié ce qui se passe.

La mécanique est en place ; reste la langue du monde qui s'est construit autour d'elle — quinze termes, que voici. Le **LLM** est un moteur ; le **mélange d'experts**, ce qui a permis de le rendre énorme sans le rendre proportionnellement coûteux à faire tourner. La **fenêtre de contexte** est sa mémoire de travail. L'**inférence**, c'est le moment où ce moteur tourne, et la **chaîne de raisonnement**, les étapes que le modèle rédige pour lui-même avant de répondre — une façon, parmi d'autres, d'obtenir une meilleure réponse en laissant tourner ce moteur plus longtemps.

Le **fine-tuning** et le **RAG** lui donnent votre savoir de deux façons bien différentes : l'un modifie le modèle lui-même, l'autre lui met les documents sous les yeux au moment de la question. La modification, en pratique, passe le plus souvent par LoRA (Low-Rank Adaptation, « adaptation de bas rang ») : on gèle le modèle et on entraîne par-dessus une mince couche de corrections — un erratum glissé dans le livre, pas une réédition. Et autour du moteur, on construit désormais des **agents** : des modèles munis d'outils, qu'on laisse travailler seuls. Le reste — tout l'outillage qui a poussé autour d'eux — porte des noms, et les cartes ci-dessous les donnent. Deux paragraphes, et vous tenez l'essentiel de l'édifice.

LE GLOSSAIRE

`LLM` — Grand modèle de langage — le moteur

Un transformer entraîné à prédire le token suivant sur d'immenses corpus de texte, puis affiné pour devenir un assistant. « Grand » renvoie au nombre de paramètres — les réglages — qui va aujourd'hui de quelques milliards à quelques milliers de milliards. Tout le reste de ce glossaire est un échafaudage construit autour d'un LLM.

« L'appli tourne sur quel LLM au juste — Claude, GPT, ou un modèle à poids ouverts ? »

`Fenêtre de contexte` — La mémoire de travail du modèle

La quantité maximale de texte (mesurée en tokens : les unités que le modèle manipule, un peu plus nombreuses que les mots) que le modèle peut considérer d'un coup. Tout ce qu'il « sait » de votre conversation vit ici ; ce qui en sort est perdu. Passée d'environ 2 000 tokens (2020) à plus d'un million (2026). À ne pas confondre avec le savoir d'entraînement : la fenêtre, c'est ce qu'il regarde, pas ce qu'il a appris.

« Le contrat fait 400 pages — ça tient encore dans la fenêtre de contexte ? »

`Inférence` — Faire tourner le modèle, par opposition à l'entraîner

L'entraînement a lieu une fois, à un coût colossal. L'inférence, c'est chaque usage qui suit : votre requête entre, des tokens sortent, un par un, chacun exigeant l'exécution du réseau entier. La facture d'électricité du secteur est de plus en plus une facture d'inférence — l'entraînement est un coût d'investissement, l'inférence un coût d'exploitation. Le calcul supplémentaire au moment de la réponse, c'est-à-dire laisser le modèle réfléchir plus longtemps sur chaque question, a placé l'inférence au centre de l'histoire du passage à l'échelle. L'anglais parle de « test-time compute ».

« Le modèle est excellent, mais les coûts d'inférence mangent notre marge. »

Mélange d'experts (MoE) — Un modèle énorme qui n'éveille qu'une fraction de lui-même

Plutôt que de faire tourner chaque paramètre pour chaque token, le réseau est découpé en de nombreux sous-réseaux spécialisés — les « experts » — et un petit routeur en choisit une poignée à chaque token. D'où les deux chiffres qu'on voit toujours cités : DeepSeek V4-Pro compte 1 600 milliards de paramètres au total, mais n'en active que 49 milliards par token. On obtient le savoir d'un modèle énorme au coût d'exécution d'un modèle bien plus petit — c'est l'astuce d'efficacité la plus importante de l'époque. Le revers : il faut tout de même stocker l'intégralité des paramètres, et donc en payer la mémoire.

« C'est un MoE de 400 Md mais seulement 17 Md actifs, donc ça tourne sur un seul nœud. »

Fine-tuning — Ré-entraîner un modèle fini pour une niche

Prendre un modèle pré-entraîné et poursuivre brièvement l'entraînement sur vos propres exemples — documents juridiques, tickets de support, style maison. Cela coûte plusieurs ordres de grandeur de moins qu'un entraînement complet, parce que le modèle connaît déjà la langue : vous enseignez une spécialité, pas la lecture.

« On a fine-tuné un modèle à poids ouverts sur les conventions de notre base de code. »

RAG — Génération augmentée par récupération — l'examen à livre ouvert

Plutôt que d'espérer que le modèle a mémorisé vos faits à l'entraînement, on va chercher les documents pertinents au moment de la question et on les colle dans la fenêtre de contexte avant qu'il réponde. L'examen à livre fermé devient un examen à livre ouvert : plus frais, vérifiable, et le remède standard au « le modèle ne connaît pas nos données internes ».

« Les réponses hallucinées ont chuté dès qu'on a mis du RAG sur la doc produit. »

Chaîne de raisonnement — Le raisonnement qu'un modèle construit avant de répondre

Plutôt que de répondre immédiatement, le modèle rédige d'abord des étapes intermédiaires — souvent longuement, souvent sans vous les montrer — et ne tranche qu'ensuite. Échanger ainsi du temps contre de la justesse s'est révélé si efficace que c'est devenu un second axe de passage à l'échelle, en parallèle de la taille du modèle. Deux réserves : réfléchir plus coûte plus cher en inférence, et la chaîne écrite n'est pas toujours le compte rendu fidèle de la façon dont le modèle est réellement parvenu à sa réponse (voir la section 04).

« Laisse-lui une chaîne de raisonnement plus longue et les erreurs de calcul disparaissent presque toutes. »

Agent — Un modèle en boucle, avec des outils

Un LLM doté d'outils (exécuter du code, naviguer, éditer des fichiers, appeler des API) et exécuté en boucle : observer la situation, agir, constater le résultat, agir encore. Le modèle est le même ; c'est la boucle qui transforme un prédicteur de texte en quelque chose qui agit. La fiabilité sur de longues chaînes d'actions est la grande bataille d'ingénierie de l'époque.

« L'agent a vu les tests échouer, lu les logs, et corrigé son propre patch. »

Harness — Le cockpit construit autour du modèle

Le logiciel qui enveloppe un LLM et en fait un agent utilisable : il gère la fenêtre de contexte, expose les outils, applique les permissions, relance après échec, décide de ce que le modèle voit et peut faire. Une grande part de la différence entre une démo spectaculaire et un produit fiable tient à l'ingénierie du harness, pas au modèle.

« Même modèle, autre harness — le jour et la nuit sur la résolution de nos tickets. »

`MCP` — Model Context Protocol — l'adaptateur universel

Un standard ouvert (introduit par Anthropic en 2024, depuis adopté par toute l'industrie) pour connecter les modèles aux outils et données externes. Avant : chaque application, pour chaque modèle, exigeait sa propre couche d'adaptation. Après : un outil parle MCP une fois, et tout modèle compatible peut s'en servir — l'USB-C de l'ère des agents.

« Expose la base de données en serveur MCP et tous les agents de la boîte pourront l'interroger. »

`CLI` — Interface en ligne de commande — là où les agents ont pris leur poste

Le terminal où les développeurs passent le plus clair de leur temps. Il est devenu le lieu de travail préféré de l'IA (Claude Code, Codex CLI, Gemini CLI) pour une raison simple : dans un terminal, tout est texte et chaque action est une commande — l'habitat naturel d'une intelligence née du texte, qui se pilote de façon bien plus fiable qu'une interface à cliquer.

« Je n'ouvre presque plus l'éditeur ; l'agent en CLI gère les refactorisations. »

`Skills` — Du savoir-faire empaqueté, chargé à la demande

Des dossiers d'instructions, de scripts et d'exemples qui enseignent à un agent une procédure répétable — « comment nous relisons les PR », « comment monter nos présentations ». Chargés seulement quand c'est pertinent, pour ne pas encombrer la fenêtre de contexte. L'expertise devient un fichier : rédigeable, versionnable, partageable en équipe.

« J'ai écrit une skill de déploiement une fois ; l'agent suit désormais notre procédure à chaque coup. »

`Vibe coding` — Programmer en décrivant, en langage naturel

Terme forgé par Andrej Karpathy début 2025 : construire un logiciel en disant à une IA ce qu'on veut et en itérant sur le résultat, parfois sans lire le code. D'une efficacité stupéfiante pour les prototypes ; controversé en production, où du code que personne n'a relu est une dette technique qui ne dit pas son nom. La version sérieuse — des ingénieurs qui dirigent des agents tout en gardant l'architecture et la relecture — est simplement la façon dont beaucoup de logiciels s'écrivent désormais.

« La démo ? Vibe-codée en un après-midi. La réécriture pour la prod a demandé une équipe, et de la relecture de code. »

`Goals` — Des objectifs durables, poursuivis sans supervision

Ce que 2026 ajoute aux agents conversationnels : donner au système un objectif persistant et vérifiable — « migre cette base de code vers TypeScript », « monte la couverture de tests au-dessus de 90 % » — et il planifie, travaille, teste sa propre production contre les critères de succès, et itère pendant des heures ou des jours, avant de revenir vers vous une fois le travail terminé — ou pour signaler qu'il est bloqué. Le mot-clé est vérifiable : les goals fonctionnent là où le succès se contrôle mécaniquement — c'est pourquoi le code (avec ses compilateurs et ses suites de tests) est le premier territoire de l'autonomie.

« J'ai posé la migration en goal vendredi ; lundi, tout était vert sur 400 fichiers. »

`AGI` — Intelligence artificielle générale — le terme que personne ne définit pareil

Une compétence de niveau humain sur l'essentiel du travail cognitif, par opposition à l'excellence sur une tâche étroite. Aucune définition ne fait consensus, et chaque annonce d'AGI est d'abord le choix d'une définition. Les seuils, eux aussi, bougent : les échecs, le go, le test de Turing et les examens de niveau doctoral ont tous fait office de test, jusqu'à ce qu'ils tombent.

« Ils annoncent l'AGI dans trois ans. » « Avec quelle définition ? »

Derrière chacun de ces mots, il y a un pari que quelqu'un a fait : publier ses modèles ou les garder ; les faire grossir ou les rendre plus efficaces ; viser des machines qui agissent ou des machines qui répondent. Reste à savoir qui a parié quoi.

---

06 LES ACTEURS

Huit laboratoires, huit paris

Le peloton de tête est fourni, et ceux qui le composent ne se ressemblent pas. Chaque grand laboratoire répond à la même question – *comment construire, contrôler et monétiser une intelligence toujours plus capable ?* – et leurs réponses divergent parce que leurs convictions divergent.

Un axe organise tout le paysage : **poids ouverts contre poids fermés**. Les poids d'un modèle sont ses réglages appris, les joyaux de la couronne. Publiez-les, et quiconque a le matériel peut les exécuter, les sonder, les affiner, prolonger votre modèle, et une fois publiés, ils le sont pour toujours : ubiquité gagnée, contrôle abandonné. Des poids ouverts ne sont pas un système ouvert pour autant : données et chaîne d'entraînement restent en général privées. Gardez-les derrière un service payant – une **API**, c'est-à-dire un accès par le réseau, sans jamais livrer le modèle lui-même – et vous conservez contrôle, garde-fous et marges. Le pari, alors : c'est la capacité qui l'emporte, pas la disponibilité.

Les laboratoires américains de pointe ont surtout choisi le fermé, Meta faisant exception ; l'écosystème chinois a massivement choisi l'ouvert, en partie parce que les sanctions en font une stratégie payante. Cette divergence – qui a choisi l'ouverture, et pourquoi – comptera peut-être davantage pour la diffusion de l'IA dans le monde que n'importe quel benchmark.

LES HUIT ACTEURS

Anthropic (San Francisco · 2021) — Poids fermés — Le pari de la sûreté d'abord

Fondée par des chercheurs partis d'OpenAI sur des désaccords de sûreté, autour d'une prémisse singulière : si l'IA puissante arrive de toute façon, la voie la plus sûre est de construire parmi les tout premiers tout en investissant plus que quiconque pour comprendre et piloter ce qu'on construit. Berceau de l'IA constitutionnelle (des modèles entraînés à respecter des principes écrits), d'une grande part de la recherche en interprétabilité de la section 04, et d'engagements de « responsible scaling » qui conditionnent le déploiement aux tests de sûreté — son nouveau palier de classe Mythos sort publiquement sous le nom de Fable 5, avec des restrictions de capacités dans les domaines à haut risque comme le cyber et la biologie.

Modèles phares · 2026 : Famille Claude 5 (Fable 5) · Claude Opus & Sonnet · Claude Code

Ce qui le distingue :

- Position dominante sur l'IA pour le code et le travail agentique (Claude Code)
- Une recherche en interprétabilité sur laquelle tout le domaine s'appuie
- Des revenus penchant vers l'entreprise et l'API plutôt que le grand public
- Publie ses cadres de sûreté même quand c'est commercialement inconfortable

OpenAI (San Francisco · 2015) — Mixte — Le pionnier du « scale and ship »

Le laboratoire qui a fait le pari de l'échelle en premier, et l'a transformé en produit grand public le plus vite adopté de l'histoire. Stratégie : atteindre le premier le plus haut niveau de capacité, déployer largement, itérer en public — en pariant que c'est le contact massif avec l'IA qui permet à la société de s'y adapter. Né comme contrepoids à but non lucratif face à Google, son évolution en géant à profit plafonné adossé à Microsoft est en soi une étude de cas sur ce que l'argent de la course en tête fait aux idéaux d'origine. En 2025, le laboratoire a de nouveau publié des modèles à poids ouverts — un geste vers l'ouverture que son nom promettait.

Modèles phares · 2026 : Série GPT-5.6 (Sol, Terra, Luna) · ChatGPT · Sora

Ce qui le distingue :

- ChatGPT : l'IA grand public par défaut, à l'échelle du milliard d'utilisateurs hebdomadaires
- Pionnier des modèles de raisonnement (série o) et de l'autonomie façon « goals » dans Codex
- Imbrication profonde avec Microsoft/Azure, plus son propre programme de datacenters Stargate
- Redéfinit à chaque sortie ce que toute l'industrie juge acceptable

Google DeepMind (Londres & Mountain View · fusion 2023) — Mixte — L'installé qui possède tout, de la puce à l'application

Le seul acteur qui possède chaque couche : le lignage de recherche (le transformer, AlphaGo, AlphaFold couronné d'un Nobel), le silicium maison (les TPU — aucune dépendance à Nvidia), les datacenters mondiaux, et la distribution via Search, Android et Workspace. Plus lent à transformer sa recherche en produits qu'elle ne le méritait — le transformer a été inventé ici et commercialisé ailleurs — mais depuis Gemini 3, il convertit son avantage structurel en sorties au rythme des meilleurs, avec la gamme Gemma pour ambassadrice à poids ouverts.

Modèles phares · 2026 : Gemini 3.1 Pro · Gemini 3.6 Flash · Gemma (ouvert) · Gemini 4 en entraînement

Ce qui le distingue :

- Intégration verticale : ses puces, son cloud, ses applis au milliard d'utilisateurs
- Un bras scientifique sans équivalent : AlphaFold a refondé la biologie structurale
- Peut casser les prix — l'inférence tourne sur son propre matériel
- Le pré-entraînement de Gemini 4 confirmé publiquement : la prochaine grande carte à retourner

Meta (Menlo Park · FAIR 2013) — Poids ouverts — La superpuissance des poids ouverts

Meta offre des poids du plus haut niveau — les modèles à mélange d'experts de Llama 4, avec une fenêtre de 10 millions de tokens — et le calcul est limpide : si l'intelligence devient gratuite, la valeur se déplace vers ceux qui tiennent le canal de distribution et l'audience — ce que Meta possède, avec trois milliards d'utilisateurs. La réorganisation mouvementée de 2025 en laboratoire de « superintelligence », avec des recrutements à neuf chiffres, signale à la fois l'ambition et les remous internes. Le cadeau, lui, est bien réel : Llama est à l'origine d'une grande part de l'écosystème ouvert mondial.

Modèles phares · 2026 : Gamme Llama 4 (Scout, Maverick) · Assistant Meta AI

Ce qui le distingue :

- Les sorties à poids ouverts occidentales les plus lourdes de conséquences
- Distribution via WhatsApp, Instagram, Facebook — aucune appli à installer
- Une flotte de GPU massive tournée vers la recherche en superintelligence
- La stratégie ouverte fait aussi office de positionnement réglementaire et de recrutement

xAI · SpaceXAI (Baie de San Francisco · 2023) — Mixte — Construire plus vite que le retard

Fondée en 2023, avec des années de retard sur ses rivaux, xAI a répondu en construisant plus vite qu'eux. Son site Colossus, à Memphis, est passé du bâtiment vide à quelque 100 000 GPU en environ quatre mois ; il en abritait près de 555 000 en 2026, sur plusieurs générations de matériel, pour une puissance de l'ordre du gigawatt. En février 2026, SpaceX a absorbé l'entreprise dans un échange d'actions la valorisant 250 milliards de dollars — la plus grosse acquisition jamais enregistrée — et l'ensemble a pris le nom de SpaceXAI, avant l'introduction en bourse record de juin 2026. Grok dispose donc de ce qu'aucun autre laboratoire n'a. D'un côté, les moyens financiers d'un constructeur de fusées, qui dispensent de lever des fonds à chaque palier. De l'autre, le réseau social X, dans lequel Grok est directement intégré : des centaines de millions de personnes y rencontrent le modèle sans rien avoir à installer ni à souscrire ailleurs. L'étape annoncée ensuite, des centres de calcul en orbite, en dit long sur ce qui bloque désormais : non plus l'accès aux puces, mais l'électricité et le refroidissement. Les GPU, eux, viennent de Nvidia comme chez presque tout le monde — H100, H200, puis GB200.

Modèles phares · 2026 : Grok 4.6 · Grok 4.5 · le centre de calcul Colossus, à Memphis

Ce qui le distingue :

- Colossus : du bâtiment vide à ~100 000 GPU en quatre mois environ
- La distribution est native — Grok est livré dans X
- Les poids de Grok-1 publiés sous Apache 2.0 en 2024 ; tout ce qui a suivi est fermé
- Le seul laboratoire détenu par un lanceur spatial, et qui compte s'en servir

Mistral (Paris · 2023) — Poids ouverts — Le champion efficace de l'Europe

Fondée par des anciens de DeepMind et de Meta, Mistral est la réponse européenne à une question lourde d'enjeux géopolitiques : quelqu'un hors des États-Unis et de la Chine peut-il construire de l'IA au plus haut niveau ? Son avantage, c'est l'efficacité — petites équipes, modèles à mélange d'experts creux qui obtiennent bien plus que leur budget de calcul ne le laisserait attendre — et la souveraineté : les entreprises et gouvernements européens qui veulent des modèles capables, exécutables sur leur propre infrastructure, sous juridiction de l'UE, achètent de plus en plus du Mistral.

Modèles phares · 2026 : Mistral Large 3 · Small 4 · nouveau MoE de pointe en accès anticipé

Ce qui le distingue :

- Le seul laboratoire européen dans le peloton de tête (ou presque)
- Des modèles phares à poids ouverts auto-hébergeables — rare à ce niveau de qualité
- Une culture d'ingénierie de l'efficacité : des résultats par GPU, pas par communiqué
- Le partenaire par défaut des initiatives européennes de souveraineté numérique

DeepSeek (Hangzhou · 2023) — Poids ouverts — Le choc d'efficacité

DeepSeek est née d'un fonds spéculatif quantitatif — un métier où l'on passe ses journées à tirer le maximum de rendement de chaque euro engagé. Le laboratoire en a gardé l'obsession : son modèle de raisonnement R1, en janvier 2025, a égalé des capacités que le marché croyait 10 à 100 fois plus chères — le choc boursier que raconte la frise de la section 01 — et a forcé chaque laboratoire à revoir ses courbes de coûts. Les contrôles américains à l'exportation la privent des puces les plus performantes. Elle en a fait une règle de conception : tirer le maximum de chaque GPU auquel elle a droit. Elle publie ses poids, et des rapports techniques d'une franchise inhabituelle.

Modèles phares · 2026 : DeepSeek V4-Pro & V4-Flash · lignée R1

Ce qui le distingue :

- A redéfini le prix qu'on croyait devoir payer pour un raisonnement de ce niveau
- Poids ouverts + articles détaillés : le monde apprend littéralement de ses méthodes
- La preuve que les contrôles à l'export ralentissent les meilleurs sans les arrêter
- V4-Pro : 1 600 Mds de paramètres mais seulement 49 Mds actifs par token ; V4-Flash : 284/13 Mds — la parcimonie comme stratégie, 1 M de contexte pour les deux

L'écosystème chinois (Alibaba (Qwen) · Moonshot (Kimi) · Zhipu (GLM) · MiniMax…) — Poids ouverts — Les poids ouverts comme grande stratégie

Derrière DeepSeek, un vivier dense : Qwen d'Alibaba (la famille de modèles ouverts qui a donné le plus de dérivés au monde), Kimi K3 de Moonshot, GLM de Zhipu, MiniMax et d'autres — adossés à des géants de la tech, recrutant dans les universités d'élite, itérant à une cadence extraordinaire. Kimi K3 pèse à lui seul 2 800 milliards de paramètres, le premier modèle à poids ouverts de la classe des 3 000 milliards : entraîné dès le départ à tolérer un stockage de très faible précision — la quantification —, il n'occupe que 1,4 To au lieu de 5,6. La logique stratégique du don des poids : faute de pouvoir acheter les meilleures puces, les laboratoires chinois jouent l'ubiquité — si les développeurs du monde construisent sur vos modèles gratuits, vous façonnez les standards, l'outillage et les flux de talents. En 2026, « open source » ne veut plus dire « second choix », et c'est largement leur œuvre.

Modèles phares · 2026 : Qwen3.6-35B-A3B · Kimi K3 (2 800 Mds) · GLM-5.2

Ce qui le distingue :

- Quatre des cinq familles de modèles ouverts les plus fortes sont chinoises
- Une capacité proche du meilleur niveau, à des prix agressifs
- Des licences réellement permissives — Qwen en Apache 2.0, GLM-5.2 en MIT — l'« ouvert » n'est donc pas ici un geste marketing
- Un effort de calcul domestique (Huawei Ascend et autres) sous contrôles à l'export
- Une pile d'IA parallèle qui s'auto-renforce — l'intrigue géopolitique de la section 07

COMMENT LIRE UN PELOTON DE TÊTE ENCOMBRÉ

Les benchmarks se dépassent chaque mois ; le positionnement est la couche stable.

Anthropic parie que la *confiance finit par payer* ; OpenAI, qu'il faut d'abord *toucher le plus de monde possible* ; Google, qu'il faut *posséder toute la chaîne* ; Meta, qu'*être déjà installé chez trois milliards de personnes* vaut mieux qu'être le meilleur ; Mistral, que la *souveraineté* a des clients ; xAI, que *construire très vite* rattrape un départ tardif ; DeepSeek, que l'*efficacité* l'emporte sur la force de frappe ; l'écosystème chinois, que *l'ubiquité des poids ouverts* finit par fixer les standards. À la prochaine annonce de modèle, ne demandez pas « est-il le meilleur ? » mais « quel pari fait-il avancer ? ». L'actualité deviendra beaucoup plus lisible.

Tous ces paris, cependant, se jouent sur la même table, et cette table est faite de silicium, de courant et de fret.

---

07 INFRASTRUCTURE & GÉOPOLITIQUE

L'intelligence s'extrait, se transporte et se défend

Le fait le plus lourd de conséquences de l'IA moderne ne se lit dans aucun article d'apprentissage automatique : le « logiciel » s'est doté, sans qu'on y prenne garde, d'une base industrielle qui rivalise avec l'industrie lourde. Et les goulots d'étranglement de sa chaîne d'approvisionnement se comptent sur les doigts d'une main.

On imagine volontiers l'IA impalpable : des modèles suspendus dans le « cloud », sans matière, présents partout et nulle part. Au contraire, c'est une industrie lourde. Derrière la moindre réponse d'un grand modèle de langage, il y a une chaîne d'usines et de machines dont chaque maillon tient à presque rien : **les graveurs à ultraviolets extrêmes d'une seule entreprise néerlandaise**, **les usines d'un seul fondeur taïwanais**, **trois fabricants de mémoire**, **un seul concepteur de GPU** ou presque, quelques campus qui avalent chacun l'électricité d'une métropole, et des réseaux à bout de souffle.

Chacun de ces maillons est un endroit où les lois de la physique, l'argent et la raison d'État se heurtent. Parcourez la chaîne :

LA CHAÎNE D'APPROVISIONNEMENT DE LA PENSÉE

Lithographie — ASML (Veldhoven, Pays-Bas)

Une puce, c'est un circuit gravé dans une plaque de silicium, et la lumière sert à en imprimer le dessin. Le plan du circuit est tracé sur un masque — une plaque de quartz, à la manière d'un pochoir. On projette de la lumière au travers, en réduisant l'image d'un facteur quatre environ, sur une plaque de silicium recouverte d'une résine photosensible. La lumière ne creuse rien : elle expose la résine, qui en garde la trace là où elle a été touchée — comme le soleil marque la peau autour d'un bracelet ; ce sont des produits chimiques qui gravent ensuite, en suivant cette empreinte. Plus la longueur d'onde employée est courte, plus les traits peuvent être fins. Les machines qui atteignent les traits les plus fins de toutes, par ultraviolet extrême (EUV), sont fabriquées par exactement une entreprise au monde : ASML. Reste à produire cette lumière, et c'est là que le procédé devient invraisemblable : la machine vaporise au laser des gouttelettes d'étain fondu, 50 000 fois par seconde, et c'est ce plasma — porté à plusieurs centaines de milliers de degrés — qui émet la lumière recherchée, d'une longueur d'onde de 13,5 nanomètres. Elle est ensuite focalisée par les miroirs les plus lisses jamais fabriqués. Chaque machine coûte des centaines de millions d'euros, s'expédie en plusieurs Boeing 747, et n'a aucun concurrent, à aucun prix.

Pourquoi c'est un goulot d'étranglement : Une seule entreprise, dans une seule ville, elle-même dépendante de fournisseurs uniques (optiques Zeiss, sources lumineuses Cymer). L'exportation d'EUV vers la Chine est interdite depuis 2019 — le goulot d'étranglement le plus serré de toute la chaîne.

Fabrication — TSMC (Hsinchu, Taïwan)

Concevoir une puce et la fabriquer sont deux industries différentes. Taiwan Semiconductor Manufacturing Company fabrique les puces de tout le monde — Nvidia, Apple, AMD — et produit environ 90 % des puces logiques les plus avancées de la planète. Une usine de pointe coûte 20 à 40 milliards de dollars et demande des années de construction ; le plus dur n'est pas de graver une puce correcte une fois, c'est de réussir des milliards de fois d'affilée. La part des puces qui sortent sans aucun défaut — le rendement — fait la différence entre une usine rentable et un gouffre, et seules des années de réglages permettent de le tenir. Ce savoir-faire s'est révélé extrêmement difficile à répliquer, même si TSMC investit désormais plus de 165 milliards de dollars dans des capacités américaines (Arizona), sous intense pression politique.

Pourquoi c'est un goulot d'étranglement : La capacité la plus avancée de l'industrie la plus critique du monde se trouve sur une île au centre de la tension sino-américaine. Ce seul fait façonne des déploiements navals, le droit de l'exportation, et des programmes de subventions à cent milliards de dollars (CHIPS Act et équivalents).

Mémoire — HBM · SK Hynix, Samsung, Micron (Corée du Sud · États-Unis)

Le calcul d'un GPU ne sert à rien si les données ne l'atteignent pas assez vite — le « mur de la mémoire ». La mémoire à haute bande passante (HBM) le résout en empilant à la verticale des puces de mémoire vive — de la RAM, comme dans un ordinateur, mais empilée et câblée à travers le silicium, à quelques millimètres du processeur. Trois entreprises seulement savent la produire en volume : les coréennes SK Hynix et Samsung, et l'américaine Micron. La génération la plus récente, HBM4, alimente les GPU Rubin de Nvidia à ~22 téraoctets par seconde, et c'est l'offre de HBM, et non celle des puces de calcul, qui limite régulièrement le nombre d'accélérateurs disponibles.

Pourquoi c'est un goulot d'étranglement : Trois fournisseurs pour toute la planète, dont deux coréens. Les règles américaines d'exportation restreignent désormais la dernière génération de HBM vers la Chine ; le chinois CXMT se hâte d'en bâtir une alternative nationale.

Accélérateurs — Nvidia (Blackwell → Rubin) (Conçus en Californie, fabriqués à Taïwan)

Le GPU a été développé pour le jeu vidéo ; ses capacités de calcul massivement parallèle en ont fait le moteur de l'apprentissage profond, et de Nvidia — brièvement l'entreprise la plus valorisée de l'histoire — le fournisseur obligé de toute la période. La génération Rubin de 2026 embarque 336 milliards de transistors et 288 Go de HBM4 par GPU, pour un débit d'inférence environ 2,8 fois supérieur à la génération précédente. Son rempart le plus solide est CUDA, la couche logicielle sur laquelle toute une industrie s'est formée. Les TPU de Google, les accélérateurs d'AMD et les puces conçues en interne par les géants du cloud lui reprennent des parts, sans menacer sa position.

Pourquoi c'est un goulot d'étranglement : Nvidia conçoit mais ne fabrique pas — chaque GPU avancé transite par TSMC et l'oligopole de la HBM. Ses puces sont l'objet premier du droit américain du contrôle des exportations, avec une politique chinoise qui oscille entre interdictions et ventes sous licence (autorisations de classe H200 en 2026).

Datacenters — Les hyperscalers (Partout où il y a du courant)

Les entraînements les plus lourds exigent des dizaines ou centaines de milliers de GPU câblés en une seule machine, de fait — des bâtiments qui se mesurent en gigawatts, parcourus de circuits de refroidissement liquide, pour des dizaines de milliards. Le programme Stargate d'OpenAI, les sites multi-gigawatts de Meta, et leurs équivalents chez Google, Microsoft, Amazon et xAI — dont le Colossus de la section précédente — sont les plus grands chantiers d'infrastructure privés de l'histoire. De plus en plus, la ressource rare n'est pas la puce : c'est le raccordement au réseau électrique.

Pourquoi c'est un goulot d'étranglement : Concentration du capital : une poignée d'acteurs au monde peuvent financer du calcul à cette échelle. C'est ce qui décide, sans que personne en débatte, de qui a le droit de concourir dans la bataille de l'IA.

Énergie — La contrainte finale (Disponible partout, et nulle part en quantité suffisante)

La consommation électrique mondiale des datacenters est projetée autour de 565 TWh en 2026 — plus que l'Allemagne en une année — pour quelque 130 gigawatts de capacité, avec des prévisions proches de 290 GW en 2030, l'IA en principal moteur. La réponse de l'industrie : des accords pour redémarrer des centrales nucléaires (Three Mile Island pour Microsoft), le financement de petits réacteurs modulaires, des turbines à gaz sur site, et des files d'attente de raccordement au réseau qui s'étirent sur des années.

Pourquoi c'est un goulot d'étranglement : L'électricité est la seule ressource qu'on ne peut ni stocker en quantité, ni faire passer en contrebande, ni rendre deux fois moins chère tous les deux ans — ce que la loi de Moore a fait pour les transistors pendant un demi-siècle. Obtenir un permis de construire une ligne à haute tension, puis la construire, prend aujourd'hui plus de temps que tout le reste : c'est ce délai-là, et non les algorithmes, qui limite la croissance de l'IA. Et c'est ici que ses ambitions se heurtent aux engagements climatiques.

Les points marquent la gravité du goulot d'étranglement : le peu d'alternatives qui existent si ce maillon casse ou vous est refusé. Rouge (●●●) : de fait, un point de défaillance unique pour la planète entière.

Une fois la chaîne sous les yeux, la géopolitique s'explique d'elle-même : il suffit de regarder où sont les verrous. Depuis 2022, Washington a fait des contrôles à l'exportation son principal levier contre les progrès chinois. Trois catégories de matériel ne peuvent plus être vendues à la Chine : les machines de gravure EUV, les accélérateurs de calcul les plus rapides, et la dernière génération de mémoire à haute bande passante (HBM). Ces interdictions ne sont tenables que parce que les quelques entreprises capables de les fabriquer se trouvent toutes aux États-Unis ou chez leurs alliés – ASML aux Pays-Bas, TSMC à Taïwan, SK Hynix et Samsung en Corée du Sud.

Le bilan est instructif, et il est double. Les plafonds sont réels : les laboratoires chinois s'entraînent sur du silicium bridé ou national, et ils le paient. Mais la contrainte a rendu ces laboratoires inventifs – les percées d'efficacité de DeepSeek sont sorties d'un laboratoire qui concevait précisément sous ces limites. Et l'embargo a précipité ce qu'il voulait empêcher : un effort d'État, méthodique, pour refaire la chaîne entière sur le sol chinois – les processeurs avec Huawei, la mémoire avec CXMT.

Washington, pendant ce temps, tangue. Interdictions, puis ventes de H200 sous licence, puis nouveaux seuils : chaque tour de vis ampute aussi les fabricants de puces américains de l'un de leurs plus grands marchés. Aucun équilibre ne s'est imposé. Une industrie qui planifiait à cinq ans travaille désormais sur un horizon politique de douze mois.

L'Europe joue une autre carte, faute d'avoir les précédentes : la règle. Son règlement sur l'IA, adopté en 2024, classe les usages par niveau de risque et impose aux modèles les plus puissants des obligations de transparence et d'évaluation. Levier réel, mais d'une nature différente : il ne crée pas de capacité, il conditionne l'accès à un marché – ce qui est exactement le pari de souveraineté que porte Mistral.

Une case, sur cet échiquier, pèse plus que toutes les autres : **Taïwan**. L'île produit l'écrasante majorité des puces de pointe. Un blocus, une guerre, et ce n'est pas l'IA seule qui se grippe – c'est l'économie numérique entière. D'où les deux lectures qu'en font les stratèges : TSMC comme « bouclier de silicium », ou comme le point de rupture unique de la puissance de calcul mondiale. Les dizaines de milliards engloutis dans les usines d'Arizona, de Dresde et de Kumamoto ressemblent au mieux à une assurance que le monde s'achète – lentement, à prix d'or, avec des années de retard sur la demande.

ÉMERVEILLEMENT ET LUCIDITÉ

Deux constats, à tenir ensemble, et tous les deux vrais. L'émerveillement d'abord : nous avons industrialisé quelque chose qui avoisine la pensée. Des machines qui raisonnent, faites de sable, de lumière et de statistique apprise, et qui progressent non pas d'année en année, mais de mois en mois.

La lucidité ensuite. Ce pouvoir tient à trois fils fragiles : une chaîne dont plusieurs maillons n'ont aucun remplaçant, une consommation électrique que les réseaux ne suivent plus, et une rivalité entre grandes puissances qui pèse sur chacun de ces maillons. Enthousiastes ou catastrophistes, qui a raison ? Ni les uns ni les autres : nous n'en sommes qu'aux prémices, et les limites comme les risques restent encore à découvrir. Et comprendre comment ces machines fonctionnent — ce que vous venez de faire — est ce qui permet de se faire une opinion qui tienne. On va justement vous demander la vôtre.

---

08 VOTRE TOUR

Une dernière prédiction

Vous avez passé sept sections sur le fonctionnement de ces machines, sur qui les construit et sur ce dont elles sont faites. La question se retourne.

Tout ce qui précède tendait à une seule démonstration : on peut se faire une opinion fondée sur ce sujet, parce que le fonctionnement de ces machines est compréhensible, et parce que cette compréhension peut changer notre façon de voir le sujet. Voici donc la question de cet essai, rendue à son lecteur. Faites votre pari, et je vous dirai le mien.

LE PARI

Dans dix ans, le mot « intelligence » appliqué à ces machines nous paraîtra…

CE QUE JE PARIE, MOI

Je pense que la question aura cessé d'être intéressante. Jusqu'au milieu du XXe siècle, un calculateur n'était pas une machine mais un métier : des gens, souvent des femmes, qui faisaient les calculs à la main pour les observatoires et les bureaux d'études aéronautiques. Plus personne aujourd'hui ne demande si un ordinateur « calcule vraiment » : le mot est passé des gens aux machines, sans que personne n'ait vraiment eu à décider. Je parie qu'« intelligence » suivra le même chemin, absorbé par l'usage, parfois abusif, que nous en ferons. Je ne dis pas que cet abus sera sans conséquence, ni que les sceptiques sur l'emploi du mot ont tort. Je dis seulement que le vocabulaire capitulera le premier ; la question, elle, restera ouverte.

Vous savez désormais comment ces systèmes fonctionnent : de quoi entrer dans le débat par la bonne porte. Le domaine aura bougé avant que vous n'ayez fini de lire – les fondations que vous venez de bâtir, elles, ne bougeront pas.

Mathematica, de David Bessis, a vulgarisé les mathématiques comme aucun autre livre ; c'est lui qui a donné envie de tenter l'exercice ici, sans prétendre en atteindre le niveau. Écrit et construit en août 2026. Les faits — noms de modèles, chiffres, données de chaîne d'approvisionnement — reflètent l'information publique à cette date ; les visualisations interactives sont des simplifications pédagogiques, signalées comme telles là où c'est pertinent. Conçu comme un essai original et écrit avec l'aide de Claude, ChatGPT et Gemini — l'idée, la direction et bon nombre de corrections restent humaines. Les vôtres sont bienvenues.