Panorama des IA

Cédric GOUY-PAILLER

Directeur de Recherche - Resp. Programme IA

CEA – Direction de la Recherche Technologique

Séminaire Aristote du 16 Novembre 2026 : Anthropologie et IA

Une analogie pour commencer



AlexNet (2012)

Trois questions pour guider la présentation :

- qu'est-ce qui est fixé ?
- qu'est-ce qui est ajusté ?
- quand la tâche est-elle spécifiée ?

1 Fixé par le concepteur

Tâche
nommer l'objet principal d'une photo, parmi 1 000 classes

Données
1,2 million de photos étiquetées à la main

Famille de systèmes (modèles)
un réseau convolutif à 8 couches

Signal
un score, élevé quand la bonne classe reçoit une forte probabilité

2 Ajusté par le calcul

Paramètres
60 millions de poids, des nombres ajustés pour augmenter le signal

Entraînement
le calcul qui ajuste ces poids : cinq à six jours sur deux cartes graphiques

3 À l'usage

Inférence
une photo en entrée, une classe en sortie ; les poids ne bougent plus

Evaluation
des photos jamais vues, aux étiquettes tenues secrètes

Sources : Krizhevsky, Sutskever et Hinton (2012).

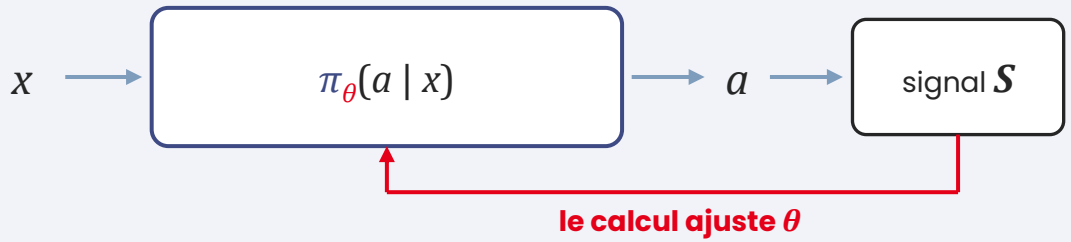
Un système d'intelligence artificielle

Ce qui est fourni

- K **Connaissances** — règles ou principes écrits
- x **Données** — des entrées collectées
- y **Étiquettes** — les sorties attendues
- sim **Simulateur** — déroule et note
- juge **Juge** — note une sortie

Entraînement

$$\theta^* = \operatorname{argmax}_\theta \mathrm{E}_{\boldsymbol{D}}[\,\boldsymbol{S}\,]$$



θ : les paramètres, des nombres. Chaque valeur de θ désigne un système π_θ.

$\boldsymbol{D}$: les données, l'expérience, les situations rencontrées pendant l'entraînement.

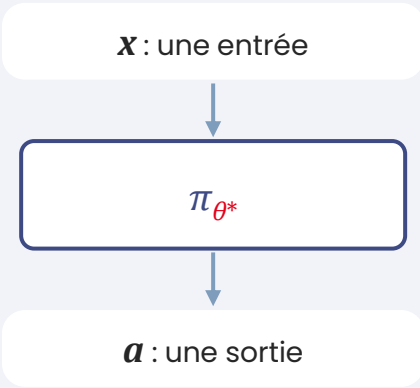
$\boldsymbol{S}$: le signal, un nombre qui note ce que π_θ produit.

$\mathrm{E}_{\boldsymbol{D}}[\,\boldsymbol{S}\,]$: la moyenne de $\boldsymbol{S}$ sur $\boldsymbol{D}$.

θ^* : la valeur de θ qui rend cette moyenne maximale (argmax).

Inférence

paramètres figés



$$a \sim \pi_{\theta^*}(\,\cdot \mid x)$$

$\pi_\theta(a \mid x)$: la probabilité que le système produise a quand il reçoit x.

$\sim$: « tiré selon ».

Écrit par le concepteur
la famille π_θ, la forme de $\boldsymbol{S}$, les algorithmes (p.ex. optimisation)

Apporté par des humains
des connaissances K, des données x, des étiquettes y

Donné par simulateur ou juge
une récompense r : un nombre qui note une sortie

Ajusté par le calcul
les paramètres θ

Système expert

Ce qui est fourni

- K **Connaissances** — Ensemble de règles
- x Données
- y Étiquettes
- sim Simulateur
- $juge$ Juge

Entraînement

$\theta^* = \text{argmax}_\theta \, E_D[\, S \,]$



Aucun entraînement : il n'y a pas de paramètre θ à ajuster.

Exemple :

SI la bactérie est à Gram négatif, en forme de bâtonnet et anaérobie, **ALORS** c'est probablement une Bacteroides (confiance 0,6).

Inférence — paramètres figés

x : le dossier d'un patient

↓

moteur$(K, \cdot)$

↓

a : un diagnostic

$a = \text{moteur}(K, x)$

moteur : le programme, écrit par le concepteur, qui enchaîne les règles de K pour répondre.

Écrit par le concepteur
le moteur d'inférence

Apporté par des humains
les règles K, écrites avec des médecins

Donné par simulateur ou juge
rien

Ajusté par le calcul
rien : aucun paramètre θ

Sources : Buchanan et Shortliffe (1984).

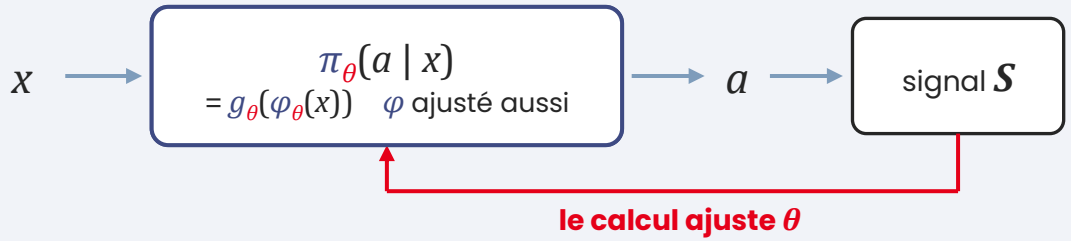
Apprentissage supervisé
Machine learning classique et réseau de neurones profond

Ce qui est fourni

- K **Connaissances**
- x **Données** — 1,2 million de photos
- y **Étiquettes** — une classe par photo
- sim **Simulateur**
- juge **Juge**

Entraînement

$$\theta^* = \operatorname{argmax}_\theta \mathrm{E}_D[\, S \,]$$



Étape	L'expérience D	Le signal S
Une étape	(x_i, y_i), $i = 1 \dots n$ une photo, sa classe	$\log \pi_\theta(y_i \mid x_i)$ **imiter** l'étiquette y_i

$\varphi_\theta(x)$: les descripteurs sont calculés par des couches dont les poids sont ajustés (AlexNet, 2012).

g_θ : la règle de décision, des descripteurs vers les probabilités de classes.

$\pi_\theta(y \mid x)$: la probabilité que le système donne à la classe y pour la photo x.

Inférence

paramètres figés

x : une photo

π_{θ^*}

a : une classe

$$a \sim \pi_{\theta^*}(\cdot \mid x)$$

En pratique, on retient la classe la plus probable.

Écrit par le concepteur
l'architecture (la forme de g et de φ), le signal S

Apporté par des humains
les photos x_i, leurs étiquettes y_i

Donné par simulateur ou juge
rien

Ajusté par le calcul
θ : tous les poids, descripteurs φ compris

Sources : Krizhevsky, Sutskever et Hinton (2012).

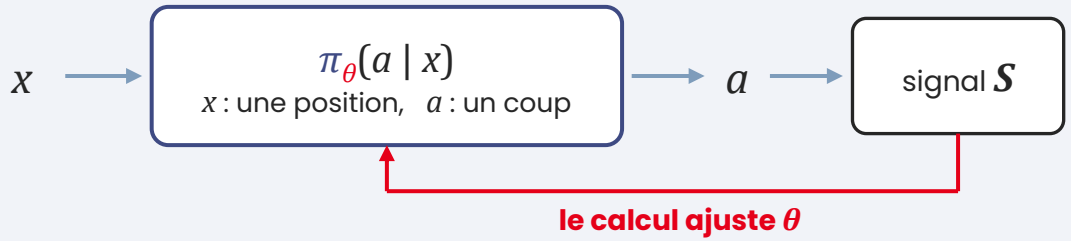
AlphaGo

Ce qui est fourni

- K **Connaissances**
- x **Données** — 30 millions de positions
- y **Étiquettes** — le coup de l'expert
- sim **Simulateur** — les règles du Go
- juge **Juge**

Entraînement

$$\theta^* = \operatorname{argmax}_\theta \mathrm{E}_D[\, S \,]$$



Étape	L'expérience D	Le signal S
Étape 1 imitation	(x_i, y_i) une position, le coup joué	$\log \pi_\theta(y_i \mid x_i)$ **imiter** le coup de l'expert
Étape 2 jeu contre soi	$\tau \sim (\pi_\theta, \mathrm{sim})$ parties contre soi-même	$r(\tau) = +1$ ou -1 **récompense** donnée par sim
Étape 3 évaluation	$(x, r(\tau))$ positions de ces parties	$-(\, v_\theta(x) - r(\tau)\,)^2$ prédire l'issue de la partie

Inférence

paramètres figés

x : une position

↓

MCTS

↓

a : un coup

$a = \mathrm{MCTS}(\pi_{\theta^*}, v_{\theta^*}, \mathrm{sim})(x)$

MCTS : une recherche écrite par le concepteur. Elle essaie des suites de coups dans sim ; π propose, v évalue.

Écrit par le concepteur
l'architecture de π et de v, la recherche MCTS, le programme sim

Apporté par des humains
30 millions de coups y_i de forts joueurs

Donné par simulateur ou juge
sim donne l'issue r : +1 gagné, -1 perdu

Ajusté par le calcul
θ : les poids de π et de v

Sources : Silver et al. (2016).

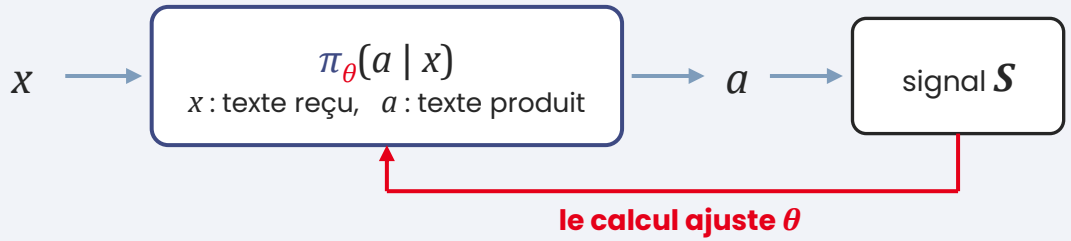
Modèle de langage (LLM)

Ce qui est fourni

- K **Connaissances** — des principes écrits
- x **Données** — textes, puis consignes
- y **Étiquettes** — des réponses rédigées
- sim **Simulateur**
- juge **Juge ou vérificateur** — ils notent la réponse

Entraînement

$$\theta^* = \operatorname{argmax}_\theta \mathrm{E}_D[\,S\,]$$



Étape	L'expérience D	Le signal S
Étape 1 pré-entraînement	$w_1, \ldots, w_T$ $w_{<t}$: le texte avant w_t	$\log \pi_\theta(w_t \mid w_{<t})$ **imiter** le mot suivant w_t
Étape 2 supervisé (SFT)	(x_i, y_i) consigne et réponse	$\log \pi_\theta(y_i \mid x_i)$ **imiter** la réponse rédigée
Étape 3 renforcement	x_i, $a \sim \pi_\theta(\cdot \mid x_i)$ a : réponse du modèle	$r = \text{juge}_K(x_i, a)$ ou $\text{vérif}(y_i, a)$ **récompense** : juge ou vérificateur

Inférence

paramètres figés

x : une consigne

π_{θ^*}

a : une réponse

$$a \sim \pi_{\theta^*}(\cdot \mid x)$$

La tâche est décrite dans x : elle est spécifiée à l'usage, par celui qui écrit la consigne.

Écrit par le concepteur
l'architecture, les signaux, le programme vérif

Apporté par des humains
les textes w, les réponses y_i, les principes K

Donné par simulateur ou juge
le juge (un modèle) ou vérif (un programme) donne la note r

Ajusté par le calcul
θ : les mêmes poids, ajustés trois fois

Sources : Brown et al. (2020) ; Ouyang et al. (2022) ; Bai et al. (2022) ; Shao et al. (2024) ; DeepSeek-AI (2025).

Introduction de connaissances humaines

Le gabarit : $\theta^* = \operatorname{argmax}_\theta \mathrm{E}_{\boldsymbol{D}}[\,\boldsymbol{S}\,]$ puis, à l'usage : $a \sim \pi_{\theta^*}(\,\cdot \mid x)$

Le système π_θ	L'expérience D	Le signal S	L'inférence
Contraindre l'architecture	**Étendre les données**	**Régulariser le signal**	**Hybrider à l'usage**
$\pi_\theta(a \mid t(x)) = \pi_\theta(a \mid x)$	$\boldsymbol{D}^+ = (t(x_i), y_i)$	$\mathrm{E}_{\boldsymbol{D}}[\,\boldsymbol{S}\,] - \lambda\,\Omega(\theta)$	$a = \mathrm{prog}(\pi_{\theta^*}, K, x)$
t : une transformation qui ne doit pas changer la réponse. L'architecture le garantit, quel que soit θ.	La même connaissance, montrée par l'exemple : on ajuste sur $\boldsymbol{D}$ et $\boldsymbol{D}^+$.	$\Omega(\theta)$: une pénalité, qui grandit quand π_θ s'écarte de ce que l'on sait. λ : son poids.	prog : un programme écrit par le concepteur. Il combine π_{θ^*} avec des connaissances K ou un simulateur.
Exemples	**Exemples**	**Exemples**	**Exemples**
Convolution : les mêmes poids en tout point de l'image.	**AlexNet :** recadrages et retournements, 2 048 fois plus d'images.	**Poids petits :** AlexNet.	**AlphaGo :** recherche et réseaux.
AlphaFold 2 : un calcul indépendant de l'orientation de la molécule.	**AlphaGeometry :** des millions de théorèmes produits par un moteur de déduction.	**Loi physique :** Ω mesure l'écart à une équation.	**AlphaGeometry :** modèle de langage et moteur de déduction, 25 problèmes d'olympiades sur 30.
		Modèle de départ : ne pas s'en éloigner (GRPO).	**RAG :** des documents ajoutés à x.

Quand la tâche est-elle spécifiée ?

La tâche est spécifiée…	Avant l'usage : conception et entraînement	À l'usage : inférence
Système expert	K **par les règles** K : conseiller un traitement antibiotique	la tâche ne change plus
Supervisé AlexNet	y **par les étiquettes** y : nommer l'objet, parmi 1 000 classes	la tâche ne change plus
Renforcement AlphaGo	r **par la récompense** r : gagner la partie de Go	la tâche ne change plus
Modèle de langage	aucune tâche unique : texte, consignes, problèmes vérifiables	x **par l'entrée** x : consigne ou exemples

Spécifier la tâche par des exemples : deux usages des mêmes (x_i, y_i)

Ajuster les paramètres : apprentissage supervisé

$$\theta^* = \text{argmax}_\theta \, \mathrm{E}_{\mathcal{D}}[\, \log \pi_\theta(y_i \mid x_i)\,] \quad \text{puis} \quad a \sim \pi_{\theta^*}(\cdot \mid x)$$

Les exemples modifient θ : un entraînement par tâche.

Écrire dans l'entrée : apprentissage en contexte

$$a \sim \pi_{\theta^*}(\cdot \mid x_1, y_1, \ldots, x_k, y_k, x)$$

θ^* ne change pas : les exemples valent le temps d'une requête.

La spécification de la tâche passe du concepteur à l'utilisateur

Défi et évaluation



Le duel

Une machine affronte un champion.

Échecs, Jeopardy, Go, poker.

Limite : une tâche unique et fermée ; la victoire ne dit rien d'un autre usage.

La tâche commune

Des données publiques, un test caché, un classement.

ImageNet, CASP, Kaggle.

Limite : 11 à 14 points perdus quand on refait le test à l'identique.

L'épreuve humaine

Un examen ou un jury conçus pour des humains.

Test de Turing, olympiades, arènes.

Limite : 73 % au test de Turing avec un rôle à jouer, 36 % sans.

Du défi au système en service

Victoire	Ce qui a suivi
Échecs, 1997	Deep Blue calcule 200 millions de positions par seconde, avec 8 000 critères écrits à la main. Quasi pas d'utilisation ensuite.
Jeopardy, 2011	IBM promet de transformer la cancérologie. Le centre MD Anderson se retire en 2017 ; la division santé est vendue en 2022, après 4 milliards de dollars investis.
Images, 2012	2016 : « il faut arrêter de former des radiologues ». Dix ans après : 10 % de radiologues en plus aux États-Unis, l'IA sert de second lecteur.
Go, 2016	Le même laboratoire se tourne vers les protéines. AlphaFold 2 (2020) : 200 millions de structures prédites, prix Nobel de chimie 2024.

Succès d'AlphaFold

1. Une tâche commune depuis 1994 : le concours CASP.
2. 170 000 structures mesurées en laboratoire, publiques.
3. Une vérification possible par l'expérience.

Trois millions de chercheurs utilisent la base aujourd'hui.

Entre la victoire et l'usage : données de terrain, place dans le travail réel, responsabilité. Le modèle n'est qu'un maillon.

General Purpose Technology?

La définition

Elle se diffuse dans tous les secteurs.

Elle s'améliore en continu.

Elle déclenche des innovations chez ceux qui l'adoptent.

Bresnahan et Trajtenberg, 1995

Le précédent : l'électricité

40 ans entre la première centrale (1882) et les gains de productivité des années 1920.

Il a fallu redessiner les usines autour de moteurs individuels.

David, 1990

L'IA générative en 2026

40 à 50 %
des actifs américains s'en servent au travail

25 %
des entreprises américaines l'emploient

+14 %
de dossiers traités en centre d'appels

Effet : encore débattu.

Réserve fédérale, 2026 ; Brynjolfsson et al., 2025

L'IA est un composant. Le gain dépend de ce qu'on réorganise autour : données, procédures, métiers.

Sources : Bresnahan et Trajtenberg (1995) ; David (1990) ; Brynjolfsson, Rock et Syverson (2021) ; Jefferson, Réserve fédérale (2026).

IA agentique

But
donné par l'humain, en langage courant

Modèle
choisit la prochaine action

Outil
recherche, code, fichiers, navigateur

Observation
le résultat revient au modèle

Résultat
rendu à l'humain

boucle répétée des dizaines de fois, sans relecture humaine

Exemple : corriger un bug informatique

1. lit le signalement
2. cherche le fichier en cause
3. modifie le code
4. lance les tests : échec
5. corrige et relance : succès
6. soumet la modification

1. **Fixé :** les outils, les droits d'accès, la boucle
2. **Ajusté :** rien pendant l'exécution
3. **Tâche :** un but en langage courant

Des dizaines d'actions s'enchaînent sans regard humain.

Conclusion

1. **Comprendre une IA.**
   Ce qui est fixé, ce qui est ajusté, le moment où la tâche est spécifiée.

2. **Performances lors d'un défi vs gain dans un système en service**
   Watson, la radiologie ; AlphaFold, données et vérification à grande échelle.

3. **LLM et agents déplacent le contrôle humain.**
   Consigne, entrées et sorties.

Un paradoxe à instruire

Jamais un outil n'a relié chacun à autant de savoirs humains.

Jamais il n'a été aussi facile de travailler sans les autres.

Chez Procter & Gamble, 776 professionnels : une personne seule avec IA fait jeu égal avec un binôme sans IA.

Dell'Acqua et al., 2025

Que devient le collectif de travail quand une personne isolée pilote des expertises concentrées ?